The <e –Adventure> Platform

Editor User Manual

Authors:

Javier Torrente Vigil

Eugenio Marchiori

Ángel del Blanco

Current Release: Version 0.9

Index

1. Getting started — *4*

1.1. Concerning <e-Adventure> — **4**

1.2. Concerning this document — **4**

1.3. Getting started: Basic projects management — **5**

2. Creating my first <e-Adventure> game — *9*

2.1. Chapters — **9**

2.2. Scenes — **10**
- 2.2.1. Adding a scene — 10
- 2.2.2. Linking scenes: adding a new exit — 14
- 2.2.3. Initial positions in a scene — 16

2.3. Items — **17**
- 2.3.1. Creating a new item — 17
- 2.3.2. Interacting with items: Actions — 18
- 2.3.3. Documentation — 19
- 2.3.4. Adding an item in a scene — 20

2.4. Set items — **21**
- 2.4.1. Create new set item — 21
- 2.4.2. Information — 23
- 2.4.3. Add set item in scene — 23

2.5. Characters — **23**
- 2.5.1. Creating a new character — 23
- 2.5.2. Documentation — 24
- 2.5.3. Add characters in scenes — 25
- 2.5.4. Conversations — 25
- 2.5.5. The player — 29

2.6. Books — **29**
- 2.6.1. Creating new paragraphs — 31
- 2.6.2. Formatted text books — 32

2.7. Cut-scenes — **35**
- 2.7.1. Slide-scenes — 35
- 2.7.2. Video-scenes — 36
- 2.7.3. Linking cut-scenes to other scenes — 37

3. Perfecting my first <e-Adventure> game: Advanced Features — *39*

3.1. Conditions and effects — **39**
- 3.1.1. A simple condition — 39
- 3.1.2. Variables — 40
- 3.1.3. Global states — 41
- 3.1.4. Either conditions blocks. — 42
- 3.1.5. What you can do with conditions — 43
- 3.1.6. Activating and deactivating flags: Effects — 43
- 3.1.7. Giving value to variables. — 44
- 3.1.8. Other effects. — 44
- 3.1.9. Macros. — 46

3.2. Organizing the elements in the scene: Layers. — **46**

3.3. Exporting the games as Learning Objects. — **48**

3.4. Exporting games as self-contained, Java ARchive files (JAR). — **51**

3.5. Adaptation and assessment profiles ____ 51
3.5.1. Assessment profiles. ____ 51
3.5.2. Adaptation profiles ____ 55
3.5.3. Set the active profile for the chapter ____ 56
3.6. Active areas ____ 57
3.7. Trajectories and barriers ____ 57
3.8. Timers ____ 60
3.9. Custom actions ____ 61
3.10. Advanced asset edition: Animations and HTML editors. ____ 62
3.10.1. Animation Editor. ____ 63
3.10.2. HTML editor. ____ 65
3.11. Advanced settings for exits. ____ 65
3.12. Other options. ____ 66

1. Getting started

1.1. Concerning <e-Adventure>

<e-Adventure> is a complete platform for the development of conventional point-and-click adventure games for educational applications, which has been developed by the <e-UCM> research group (http://www.e-ucm.es) of the Complutense University of Madrid. When we refer to adventure games we bear in mind titles such as the *MonkeyIsland*TM or *Myst*TM sagas. This genre is one of the most suitable for education due to its intrinsic features; hence the choice.

The current release of the platform is version 0.9, which can be downloaded from http://e-adventure.e-ucm.es/. In this web page you will find more information and sample games for you to try.

There are various downloadable versions of <e-Adventure>. Three are platform-dependent (Windows32, Linux & MAC Leopard) and another one is multiplatform. We recommend using the platform-dependent versions, but anyway the multiplatform version should work fine if you have installed Java 6 in your computer (JRE 1.6.0_02 recommended). Besides, we are dedicating our best efforts to improve our compatibility with MAC OS, but it is still under testing. We apologize for all the inconveniences and problems you could find in the MAC version.

All the downloadable versions of <e-Adventure> located on the web page are mainly a .zip file containing the editor and the engine files. The editor is the application you need to produce the games. The engine is what you need to use in order to play the games. Both applications are compiled as java jars; hence you only need to uncompress the files and execute the jar contained in it. Moreover, you will find two bat files for the execution of the applications under Windows OS and .sh files as shortcuts for MAC and Linux. Please, do not remove or move any of the files contained in the main folder of the release, as they need to be in the same folder. Otherwise the behavior of the applications will be unexpected.

In the main directory of <e-Adventure> (EAD_HOME thereafter) three folders deserve your attention. Those are *Projects*, *Exports* & *Reports*. The *Projects* folder will contain by default of the adventure projects you create with the editor. The second one, *Exports*, will allocate by default the executable versions of those projects that you create. Finally the *Reports* folder will contain the assessment reports produced when the games are played using the game engine.

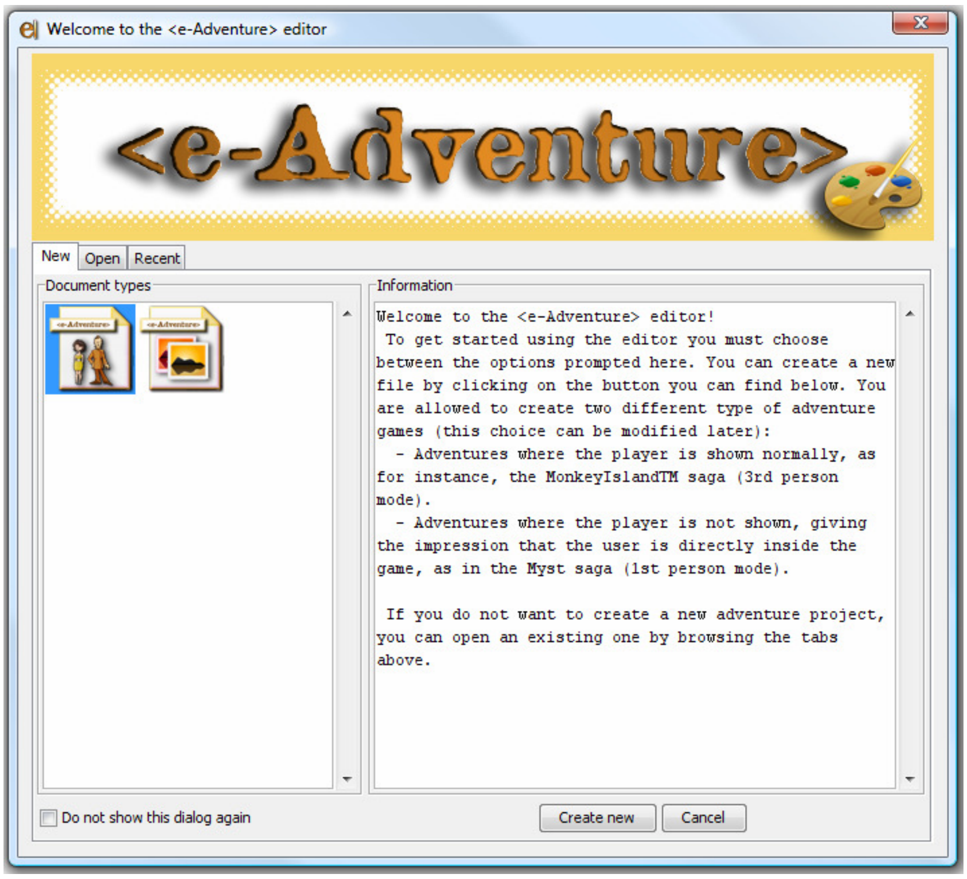
Please, before using the <e-Adventure> platform take some minutes to update your java JRE (Java Runtime Environment) version (http://www.java.com). The release 0.9 has been compiled for the 1.6 version of the JRE For older versions of the JRE we cannot guarantee the applications will work properly (we specially recommend JRE 1.6.0_02, download it here)). In addition, if you find this project interesting you can find more information about our publications and other initiatives on our research group's Web site (http://www.e-ucm.es), such as the <e-Adventure3D> project (http://e-adventure3d.e-ucm.es).

1.2. Concerning this document

Hello and welcome to the <e-Adventure> platform. In this tutorial you will learn how to produce and execute a complete functional game using our adventure editor. It is not its aim to provide a full detailed document covering all the aspects of both <e-Adventure> editor and game engine, but a brief guide to introduce a novice user in the development of an adventure game from the beginning to the end. More documentation will come soon.

1.3. Getting started: Basic projects management

Now you have been introduced to the platform, let's get started. Firstly we will get in touch with the adventure editor. Once you have executed the application, you will see a dialog like the following:



```
Welcome to the <e-Adventure> editor!
 To get started using the editor you must choose
between the options prompted here. You can create a new
file by clicking on the button you can find below. You
are allowed to create two different type of adventure
games (this choice can be modified later):
  - Adventures where the player is shown normally, as
for instance, the MonkeyIslandTM saga (3rd person
mode).
  - Adventures where the player is not shown, giving
the impression that the user is directly inside the
game, as in the Myst saga (1st person mode).

 If you do not want to create a new adventure project,
you can open an existing one by browsing the tabs
above.
```

Through this dialog you will be able to create new adventure projects or open existing projects.

Creating a new project.

Since <e-Adventure> 0.5 the games are managed as projects, which are just folders containing all the assets of the game. To get the games running in the game engine firstly we will need to export the projects as .ead files.

To create a new project you just need to press the **"Create new"** button. Then you will be prompted to introduce the name of the root folder of the project and its parent folder (by default this folder will be *Projects*, a folder contained in the EAD_HOME). You must take into account that the name of the folder can only contain numbers, characters blanks and the following symbols: [,] , (,), _, -. Any other characters, such as /, |, @, \, & o $ are completely forbidden in the name of the projects.

Two different types of projects can be created with the editor, each represented by a graphic motif in the dialog. Those will lead to the creation of adventure games, properly talking.

Why two different icons for an adventure game? The answer is easy. <e-Adventure> is devised for the development of two kinds of games. For simplicity, from now on we will refer to both types as *MonkeyIsland*-like games, which will be named "Third Person Adventures" and *Myst*-like games, which will be named "First Person Adventures". The main characteristics of both types are:

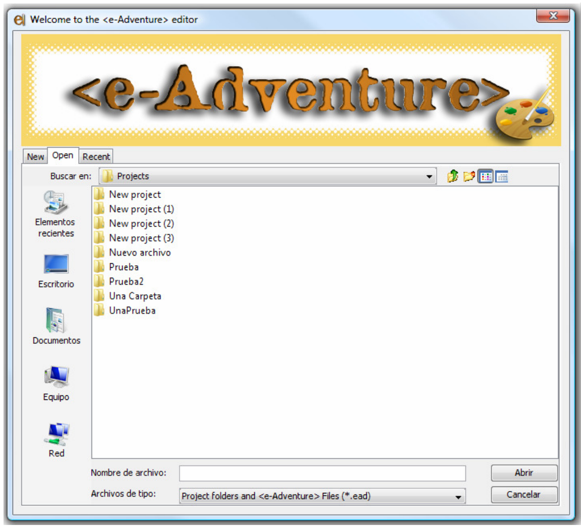
Third Person Adventures. The player is represented by an avatar, which is drawn onto the game all the time. It needs some time to go from place to place, and when he speaks the text is displayed just over his head.

First Person Adventures. There is no avatar for the player. The player explores the game himself, and transitions between scenes are instantaneous. Usually these games are designed using photos to configure the scenes. Hence the player interacts in first person with a very real-looking world, in which you can turn or go back in the scene bi clicking left, right, down, etc.

Loading an existing project, an .ead file or a recent project.

You can use the start dialog to find and load files and projects. Besides you will find a list of recent projects for your convenience. By default the Open Dialog will be pointing to *Projects* which is in the EAD_HOME. This folder is devised to hold all the projects, but you can place them where you want in your system (however you must always consider the name restrictions aforementioned).

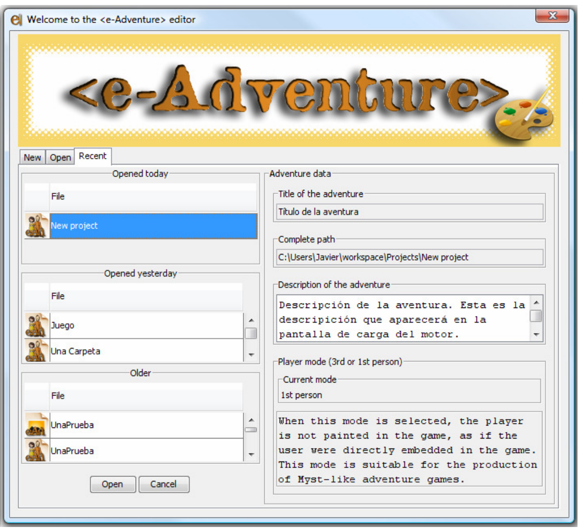
To open a project you will need to select the folder of the project and press on "Open". Besides, this tab can be used to import .ead files as a new project.



Buscar en: Projects

New project
New project (1)
New project (2)
New project (3)
Nuevo archivo
Prueba
Prueba2
Una Carpeta
UnaPrueba

Nombre de archivo:
Archivos de tipo: Project folders and <e-Adventure> Files (*.ead)



Opened today
File
New project

Opened yesterday
File
Juego
Una Carpeta

Older
File
UnaPrueba
UnaPrueba

Adventure data

Title of the adventure: Título de la aventura

Complete path: C:\Users\Javier\workspace\Projects\New project

Description of the adventure: Descripción de la aventura. Esta es la descripición que aparecerá en la pantalla de carga del motor.

Player mode (3rd or 1st person)
Current mode: 1st person

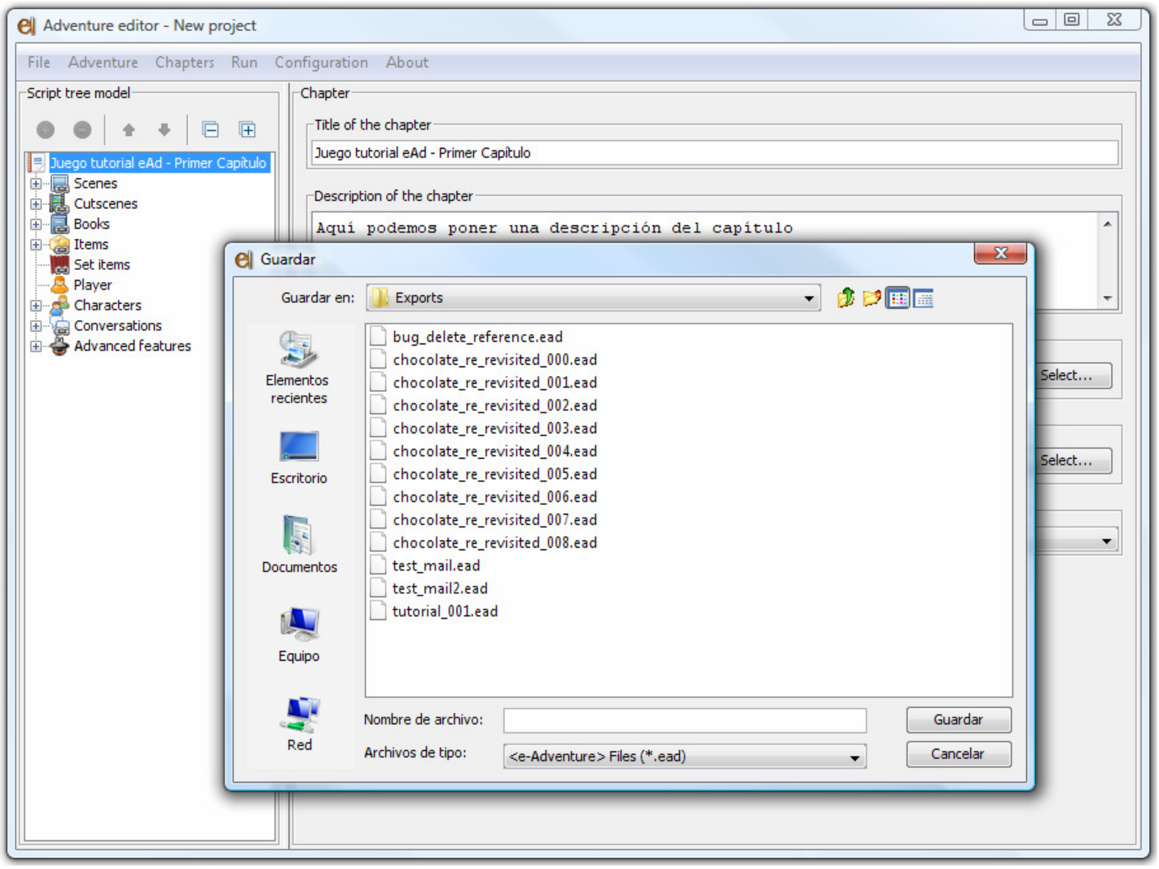
When this mode is selected, the player is not painted in the game, as if the user were directly embedded in the game. This mode is suitable for the production of Myst-like adventure games.

Exporting a project as an .ead file

As it has been previously mentioned, the basic work unit in the editor is the *Project*. However, a project may contain hundreds of files, which can harden the delivery of the games to the students. Thus the game engine loads the games as *ead* files, which are just simple zip files containing all the resources of the adventure. Therefore you need to follow the next steps in order to export a project as an ead file:

1) Check the adventure consistency. Go to the adventure menu on the Windows menu and press the option "Check adventure consistency". If the adventure is consistent (i.e. there are no mistakes impeding that the engine could read the game) you can proceed with the exportation. Otherwise you will need first to solve the problems encountered.

2) Export the game as .EAD file. Follow the next process: *File Menu > Export project > Export as EAD file*. Then type how you want to name the file and select the destination folder, and after a while (be patient, the process can take a bit long) the game will be ready to be loaded with the game engine. By default the parent folder of the file will be *Exports,* located in the EAD_HOME.



Title of the chapter: Juego tutorial eAd - Primer Capitulo

Description of the chapter: Aquí podemos poner una descripción del capitulo

Guardar en: Exports

bug_delete_reference.ead
chocolate_re_revisited_000.ead
chocolate_re_revisited_001.ead
chocolate_re_revisited_002.ead
chocolate_re_revisited_003.ead
chocolate_re_revisited_004.ead
chocolate_re_revisited_005.ead
chocolate_re_revisited_006.ead
chocolate_re_revisited_007.ead
chocolate_re_revisited_008.ead
test_mail.ead
test_mail2.ead
tutorial_001.ead

Nombre de archivo:
Archivos de tipo: <e-Adventure> Files (*.ead)

Importing an .ead file as a new project

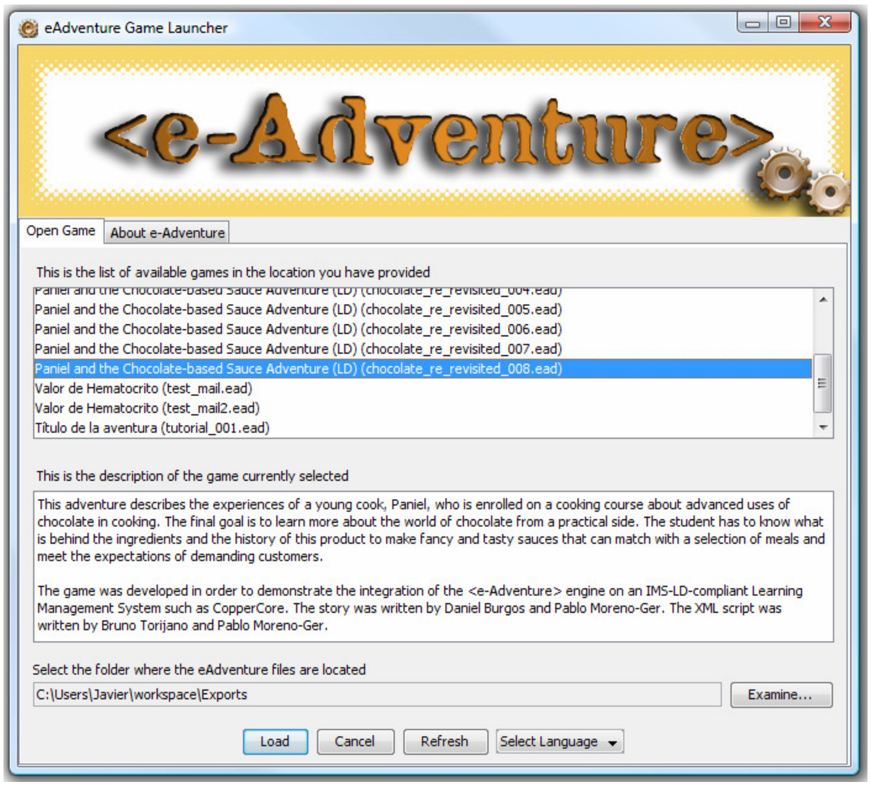
You can also import .EAD files as new projects. Just follow the next steps: *File menu > Import > Import game as new project.*

Use the file dialog that will appear on the screen to select the .EAD file to import. After that you will see another one to select the folder of the new project. Then the game will be decompressed in the folder and the project will be loaded.

Running an adventure name with the game engine

As it has been previously mentioned the <e-Adventure> game engine can only load the games as EAD files. To get the game engine running you just need to follow a process which is completely analogous to the execution of the game editor. You will get a dialog like the next one. This dialog allows us to explore folders and select the adventure we want to launch.

By default the contents of the *Exports* folder will be shown on the screen for convenience of the game authors. To change the current folder use the button "Examine" and use the file chooser dialog that will appear on the screen to select the folder to explore or the EAD file to run.



This is the list of available games in the location you have provided

Paniel and the Chocolate-based Sauce Adventure (LD) (chocolate_re_revisited_004.ead)
Paniel and the Chocolate-based Sauce Adventure (LD) (chocolate_re_revisited_005.ead)
Paniel and the Chocolate-based Sauce Adventure (LD) (chocolate_re_revisited_006.ead)
Paniel and the Chocolate-based Sauce Adventure (LD) (chocolate_re_revisited_007.ead)
Paniel and the Chocolate-based Sauce Adventure (LD) (chocolate_re_revisited_008.ead)
Valor de Hematocrito (test_mail.ead)
Valor de Hematocrito (test_mail2.ead)
Titulo de la aventura (tutorial_001.ead)

This is the description of the game currently selected

This adventure describes the experiences of a young cook, Paniel, who is enrolled on a cooking course about advanced uses of chocolate in cooking. The final goal is to learn more about the world of chocolate from a practical side. The student has to know what is behind the ingredients and the history of this product to make fancy and tasty sauces that can match with a selection of meals and meet the expectations of demanding customers.

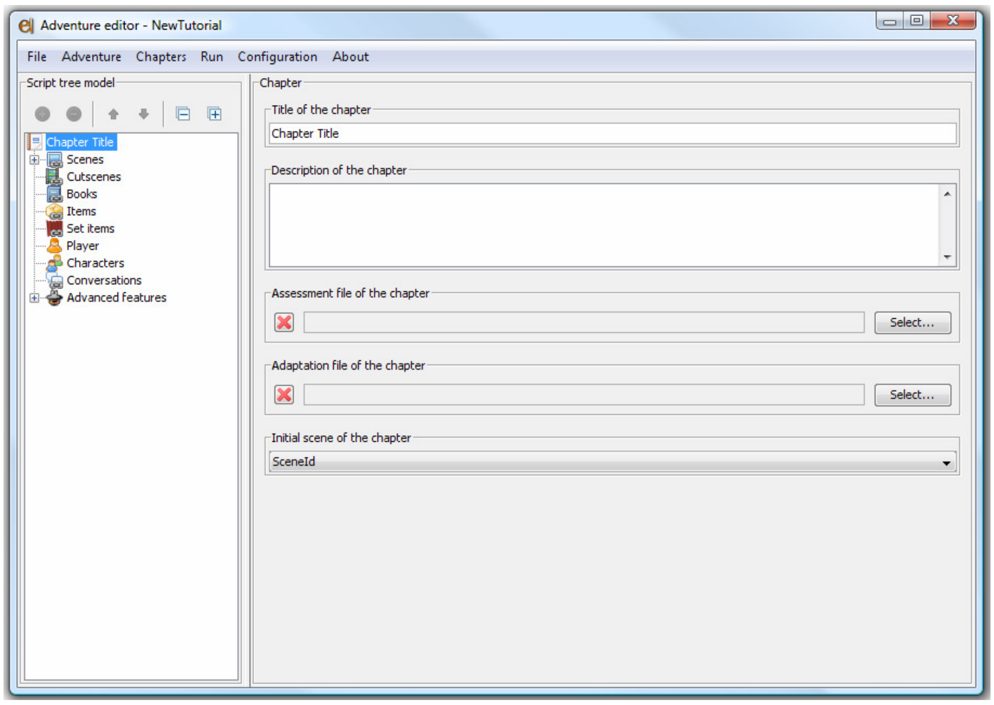
The game was developed in order to demonstrate the integration of the <e-Adventure> engine on an IMS-LD-compliant Learning Management System such as CopperCore. The story was written by Daniel Burgos and Pablo Moreno-Ger. The XML script was written by Bruno Torijano and Pablo Moreno-Ger.

Select the folder where the eAdventure files are located

C:\Users\Javier\workspace\Exports

2. Creating my first <e-Adventure> game

Once you have started the application, you will see the next picture. First of all let's have a look on what you can do here. The adventure editor is organized following a simple structure. On the left there is a data tree, which is used to index all the elements that can be defined in a game. When you click on a node of the tree on the right the options you can edit are displayed in a panel. Therefore you can use the left tree to add, remove, access and organize the elements of the adventure. You can perform those operations you can use the tree menu on top or by right-clicking on the element. Then, when you select an element you can edit it on the right panel. On top of both, the menu bar allows you to edit other aspects of the game, and will be later explained. First, let's explain the elements conforming an <e-Adventure> game.


Adventure editor - NewTutorial
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
Cutscenes
Books
Items
Set items
Player
Characters
Conversations
Advanced features
Chapter
Title of the chapter
Chapter Title
Description of the chapter
Assessment file of the chapter
Select...
Adaptation file of the chapter
Select...
Initial scene of the chapter
SceneId

2.1. Chapters

An <e-Adventure> game is organized in chapters. In this manner the adventure is fragmented in small pieces easy to design, edit, and keep in memory when executing. Each chapter is a self-contained mini-game. All the elements defined in a chapter cannot be accessed in others.

In the editor, only one chapter is edited at the same time. The "chapters" menu on top allows you to add and remove chapters, and browse the chapter you would like to edit. Through this menu you can manage the flags of the current chapter, but this will be discussed later.

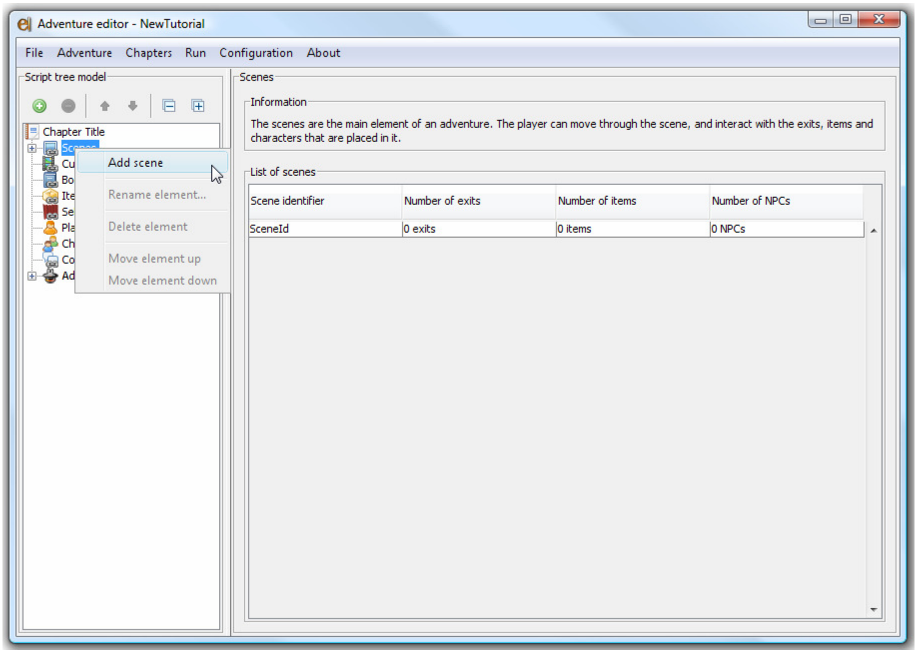
When a chapter has already been selected, you can edit its title, a description, and the initial scene of the chapter. Besides, you can select the active assessment and adaptation profiles for the chapter, as the previous figure depicts. You will learn to do that when we get to the "Adaptation and Assessment" section.

2.2. Scenes

Each chapter is organized in scenes. There are two main types of scenes: conventional scenes and cut-scenes. The conventional scenes are scenarios where the player interacts with objects and characters, and are linked with other scenes through exits. On the other hand, cut-scenes are vehicles to enhance the educational value of the adventures. Fundamentally, two types of cut-scenes are supported: slide-scenes and video-scenes. The first one is made of a succession of slides, which are full-screen images displayed one after the other. A video-scene is, as its name indicates, a video which is played on the screen. In this section will learn to produce new scenes and cut-scenes.

2.2.1. Adding a scene

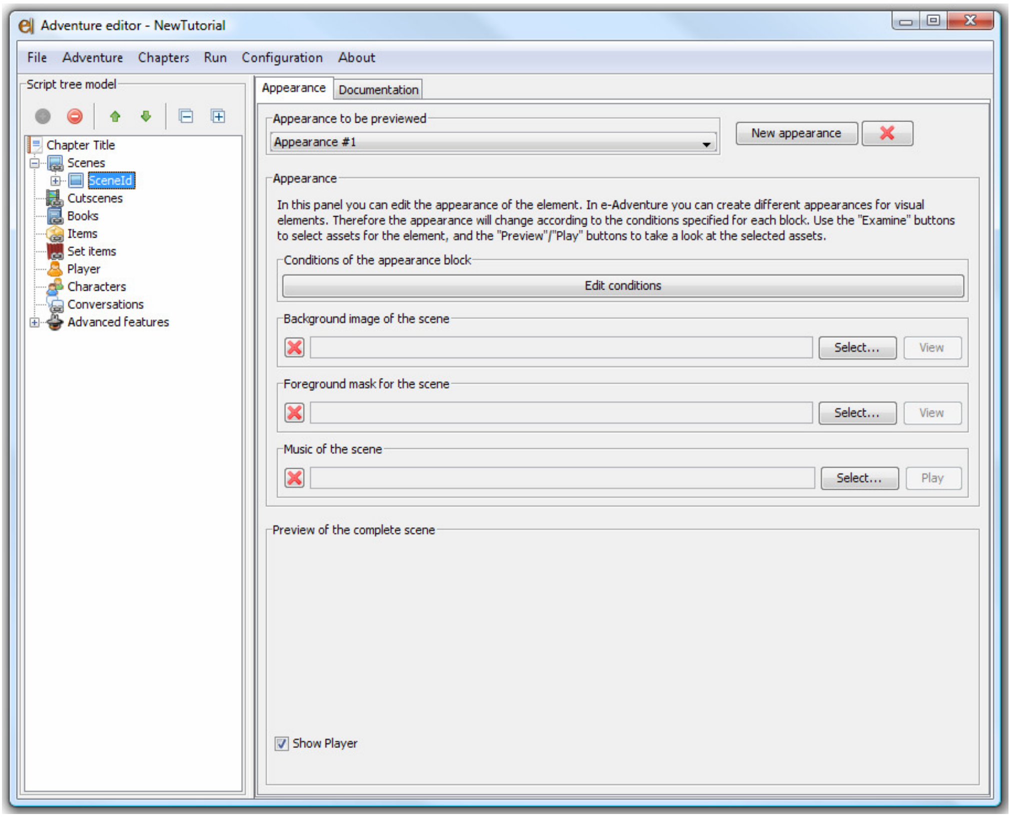
First, you need to click with the right button of the mouse on the node called "Scenes" or "Cutscenes" in the left tree, depending of the type of scene we want to create, as the figure depicts.



The scenes are the main element of an adventure. The player can move through the scene, and interact with the exits, items and characters that are placed in it.

Scene identifier	Number of exits	Number of items	Number of NPCs
SceneId	0 exits	0 items	0 NPCs

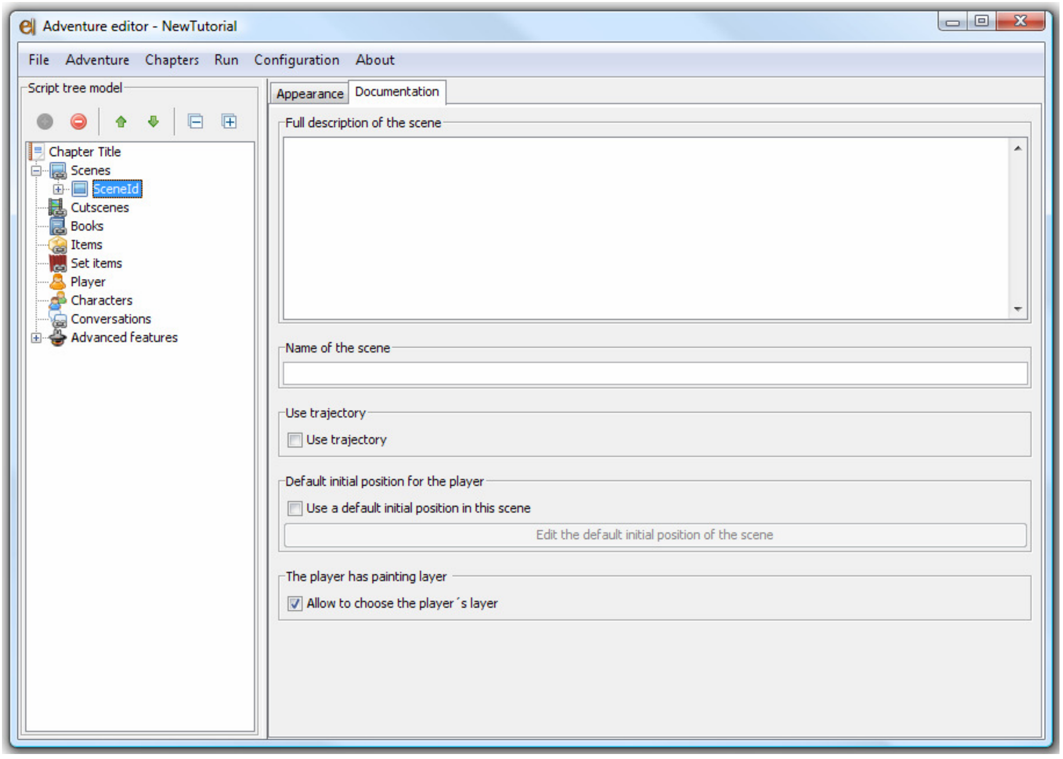
Let's start creating a new conventional scene. Once a new scene has been created, we are ready to edit it.



In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

Conditions of the appearance block
Edit conditions
Background image of the scene
Select...
View
Foreground mask for the scene
Select...
View
Music of the scene
Select...
Play
Preview of the complete scene
Show Player

As you can see, for scenes two tabs are available. The "documentation" tab is available for all the elements in <e-Adventure>. In this case it admits the edition of a full description of the scene, its name. It is important no to mistake the identifier and the name of the elements (which is their name on the left tree). The name has any restrictions but the identifier must start by a character and cannot contain blanks.


Adventure editor - NewTutorial
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
SceneId
Cutscenes
Books
Items
Set items
Player
Characters
Conversations
Advanced features
Appearance
Documentation
Full description of the scene
Name of the scene
Use trajectory
Use trajectory
Default initial position for the player
Use a default initial position in this scene
Edit the default initial position of the scene
The player has painting layer
Allow to choose the player´s layer

Besides, there are advanced options that can be edited here, as the initial position of the player in this scene. However, we will come back to this point when the advanced options of the scenes are described.

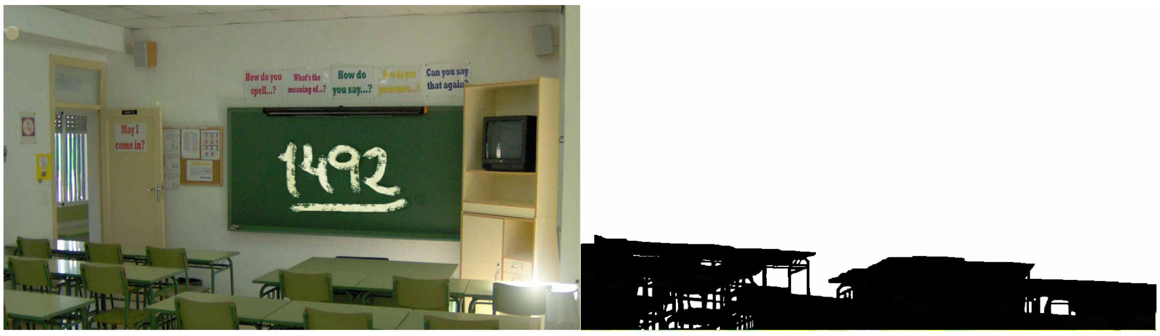
The "appearance" tab provides you with features to select what the scene looks like, which depends on four assets. Only one of them is compulsory (background image), the others are completely optional:

Background image (compulsory)

It is the most importance, as will be always drawn in the scene. Background images must be, at least, 800x600. In <e-Adventure> the next types of images are supported: PNG, JPEG (JPG), BMP, WBMP and GIF (with no animation).

Foreground mask (optional)

Black & white image which is used to know what part of the background image must be drawn behind the characters and player and which must be rendered before. Next there is an example of how a foreground mask works.



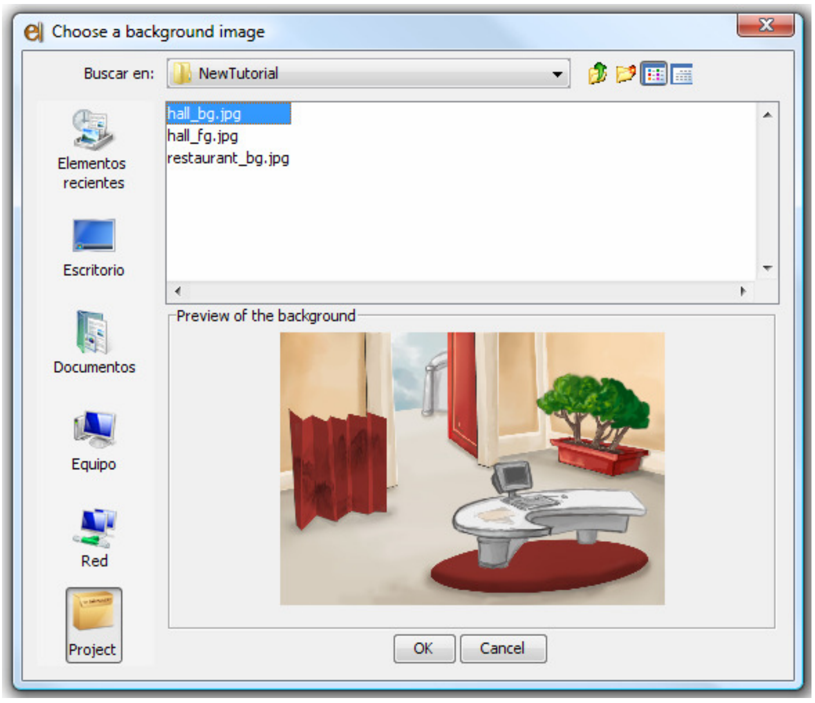
The left figure depicts a classroom. The ideal behaviour is that player and characters will be drawn between the desks and the rest of the image. Thus the mask will be an image where the desks (foreground) are drawn completely in black and the background in white.

Foreground masks are beginning to be deprecated in <e-Adventure> because they are difficult to use and they show clear limitations. As this document describes the same functionality can be obtained combining *set items* and *layers*.

Music (optional)

The background music of the scene, which will be played all the time in a loop. MP3 and MIDI formats are accepted.

For instance, we will add background and foreground to the scene. To add both assets we just need to press the select button. Then a chooser utility will be displayed:

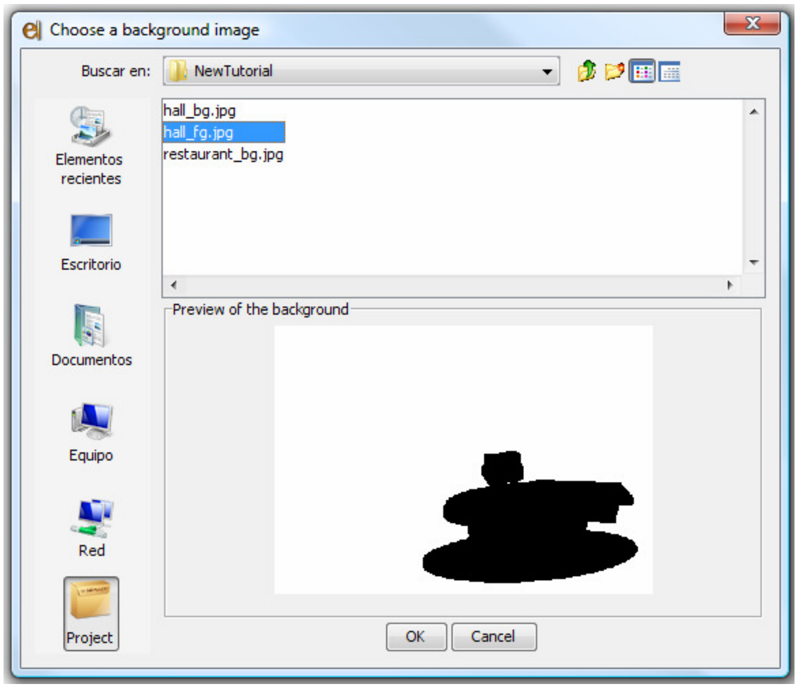


hall_bg.jpg
hall_fg.jpg
restaurant_bg.jpg

The chooser allows you to select a valid asset. Remember only 800x600 images are valid for background assets. On the bottom the asset is previewed. When you select an asset and press ok, this will be automatically added to your adventure file.

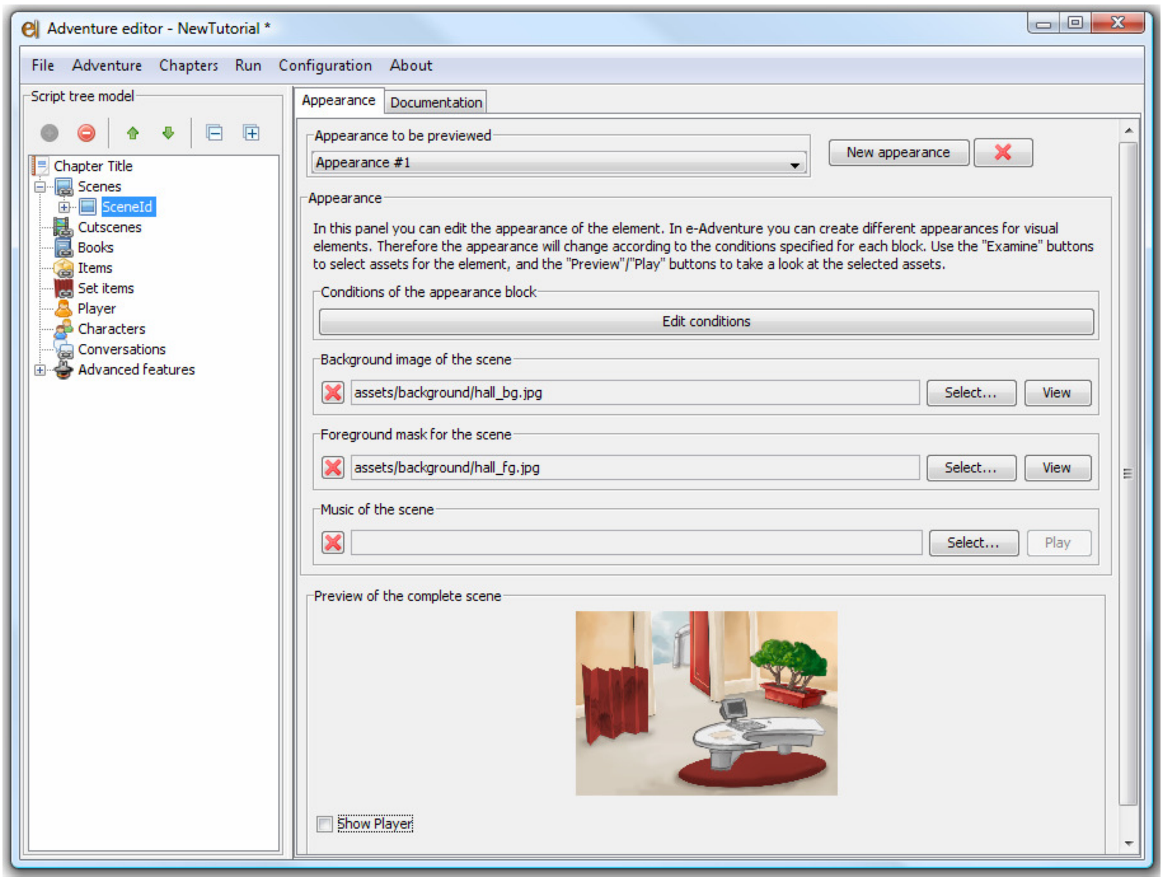
There is a shortcut for you to access the assets you have already used. If you press on the ***Project button*** on your left the assets of that category already included in the game will be displayed.

Once the background image has been added we are ready to add the foreground mask. As you see, the selected background has a computer desk at the front. When the player or a character moves to that part of the scene we would like it to be drawn behind this object, so we insert a foreground mask:



hall_bg.jpg
hall_fg.jpg
restaurant_bg.jpg

After that, the scene is previewed in the panel:


Adventure editor - NewTutorial *
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
SceneId
Cutscenes
Books
Items
Set items
Player
Characters
Conversations
Advanced features
Appearance
Documentation
Appearance to be previewed
Appearance #1
New appearance
Appearance

In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

Conditions of the appearance block
Edit conditions
Background image of the scene
assets/background/hall_bg.jpg
Select...
View
Foreground mask for the scene
assets/background/hall_fg.jpg
Select...
View
Music of the scene
Select...
Play
Preview of the complete scene
Show Player

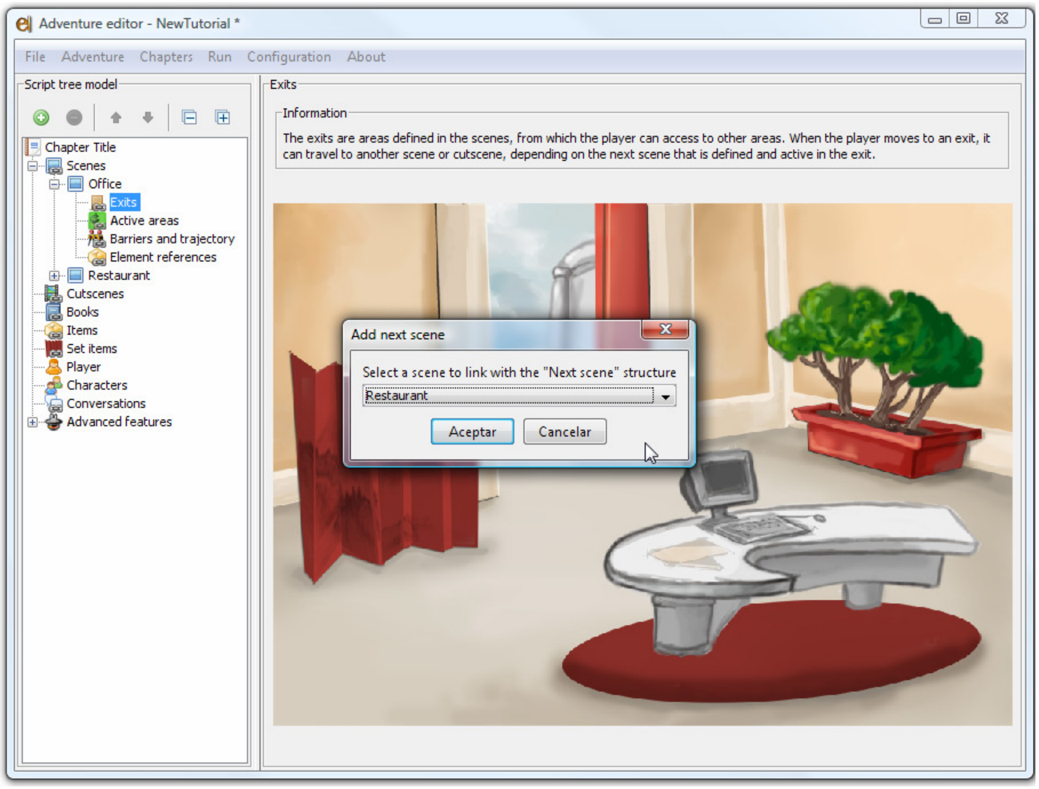
In the scene you can also define *exits, active areas, barriers and trajectories and references* to *items, set items* or *characters*. Exits are just simple transparent rectangles, that when are pressed with the mouse leads to another scenes. We will take up again this point later on when we know how to create items, set items and characters. For now, let's see how to create an exit to link the scene with other scenes.

2.2.2. Linking scenes: adding a new exit

We need to create exits if we want scenes to be connected. Otherwise the player will be trapped in the same scenario all the time.

Suppose we have two scenes; the one we created in the last section, which have been renamed to "Office" and a new one, called "Restaurant". As you see, our office has a door which is opened. We can use that graphical feature to link the scene with the restaurant, so when the player clicks there moves automatically to the restaurant.

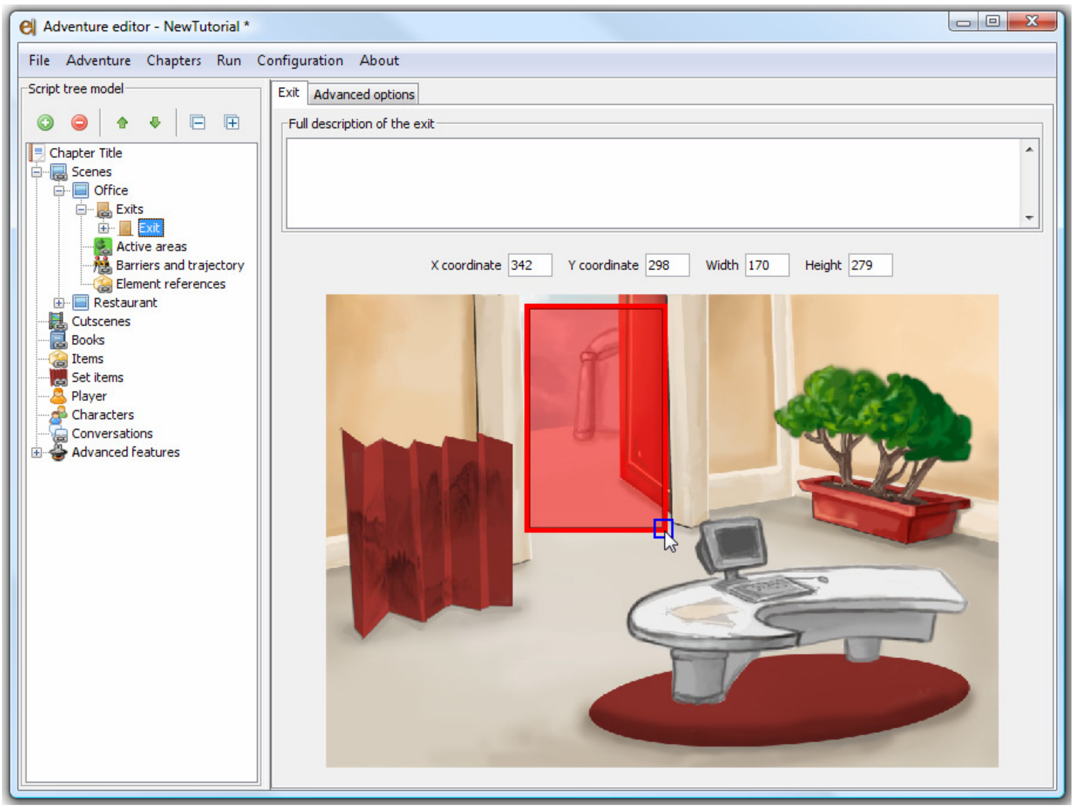
The first step is to press on the node "Exits" which hangs from the node of the scene. Perform a right click and select "Add exit". Then you will be prompted to choose which scene you want the player to move to. In our case it is the scene "Restaurant".



The exits are areas defined in the scenes, from which the player can access to other areas. When the player moves to an exit, it can travel to another scene or cutscene, depending on the next scene that is defined and active in the exit.

Select a scene to link with the "Next scene" structure

The next panel you see will allow you to select the area the exit will occupy. You just need to draw a rectangle with the mouse, which will be red-drawn. Do not worry, in the game the exit will be completely transparent; here is given a color so you can know what you are doing.



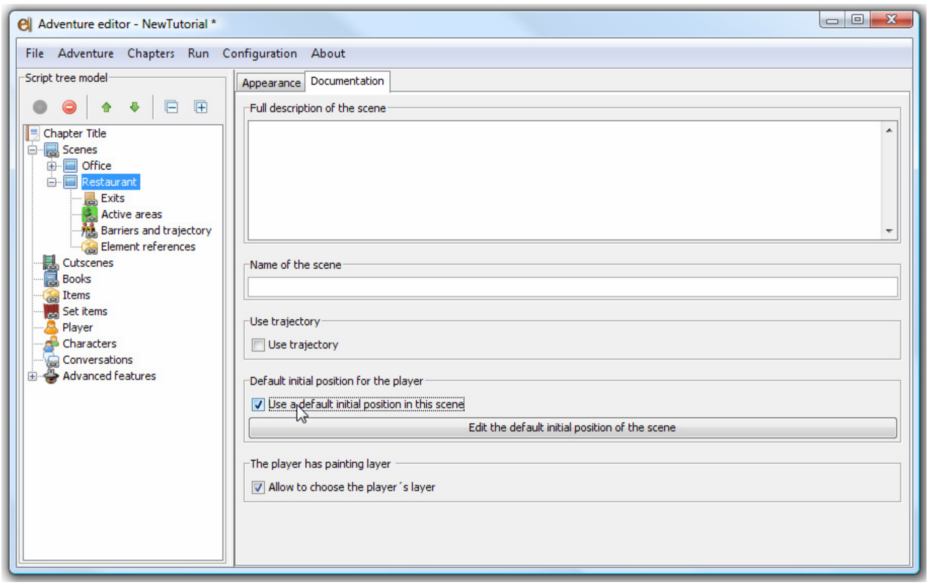
X coordinate 342 Y coordinate 298 Width 170 Height 279

Then, when the player points the mouse over this rectangle, the name of the next scene ("Restaurant") will be displayed. After a click the player will be situated in the scene "Restaurant".

2.2.3. Initial positions in a scene

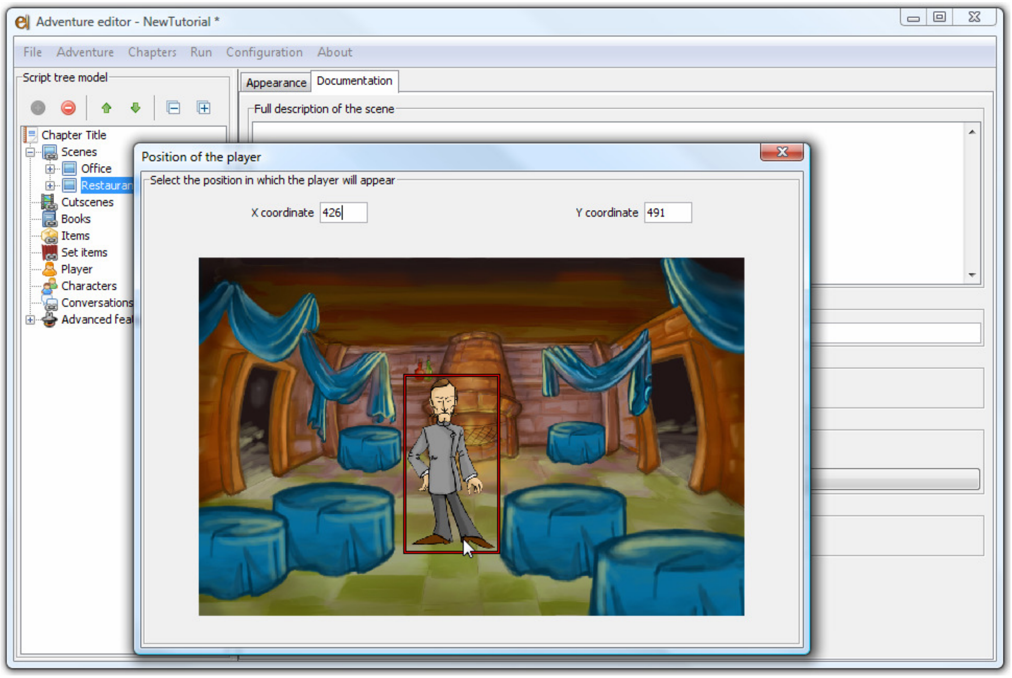
If you execute the game we have created so far, the player would move from the office to the restaurant through the exit. When the player appeared in the new scene, it will be located "wherever the game wants". To avoid that disgusting effect, we are going to modify the initial position of the restaurant scene. You can use the checkbox and the button on the bottom of the documentation panel of the scene to get a preview of the scene and select the initial position of the player by simply clicking on it.

The first step is get the "Restaurant" node selected on the left tree. Then go to the documentation tab and activate the check box "Use a default initial position in the scene". Then click on the button below.



Then you will see a preview of the scene. Click on the position you want the player to appear on and the player will be drawn there each time the scene is rendered.

The next image depicts that, assuming you have already edited the player, which in our case is a chef:

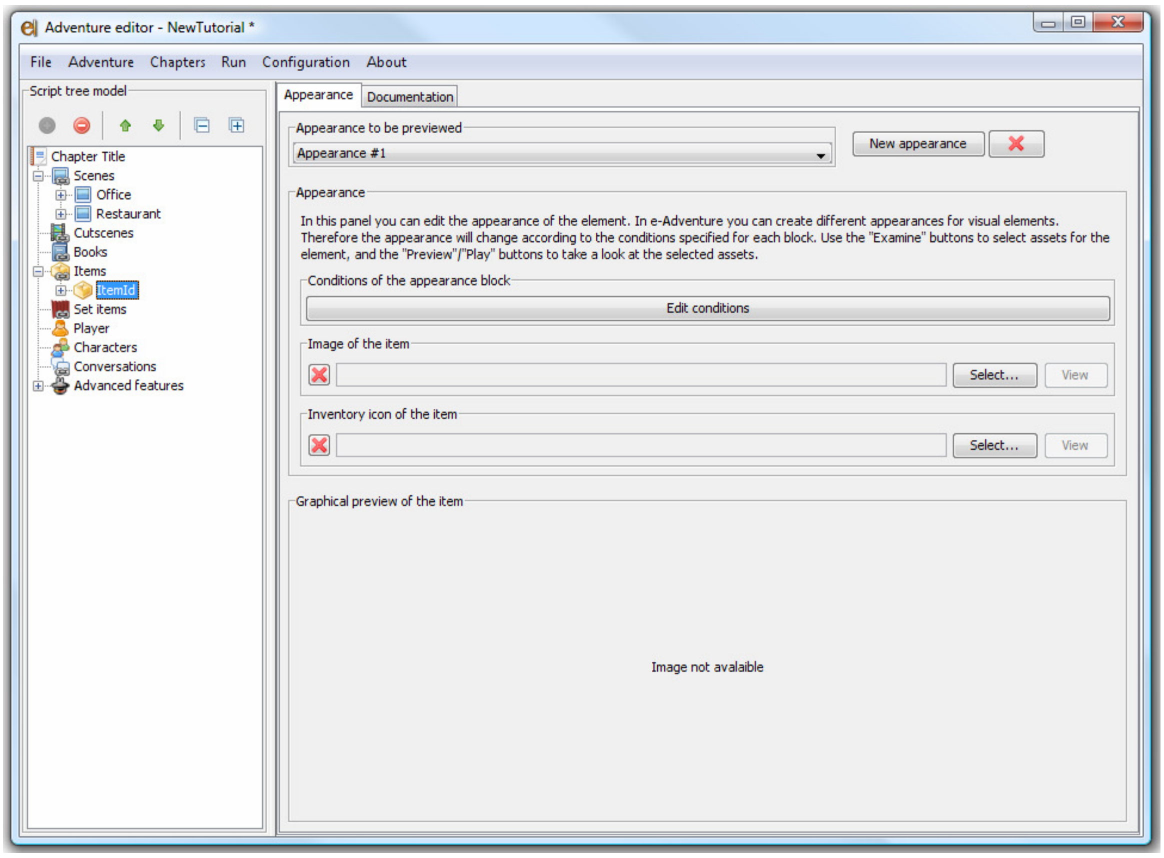


2.3. Items

In <e-Adventure> games you can define objects for the player to interact with them. These objects are called "items".

2.3.1. Creating a new item

Creating a new item is as simple as creating a new scene. First of all you must right-click on the node labeled as "items" on the left panel. Then press the option "Add item". You will be prompted to type the id for the item. Every single element in <e-Adventure> is given a unique id. This is used to correctly identify each one, as you may refer them from different places. Remember those ids must not contain blank spaces, and must be unique. Otherwise an error will be reported. Then you will get a screen like the one below, which is very similar to the one we got creating a scene:

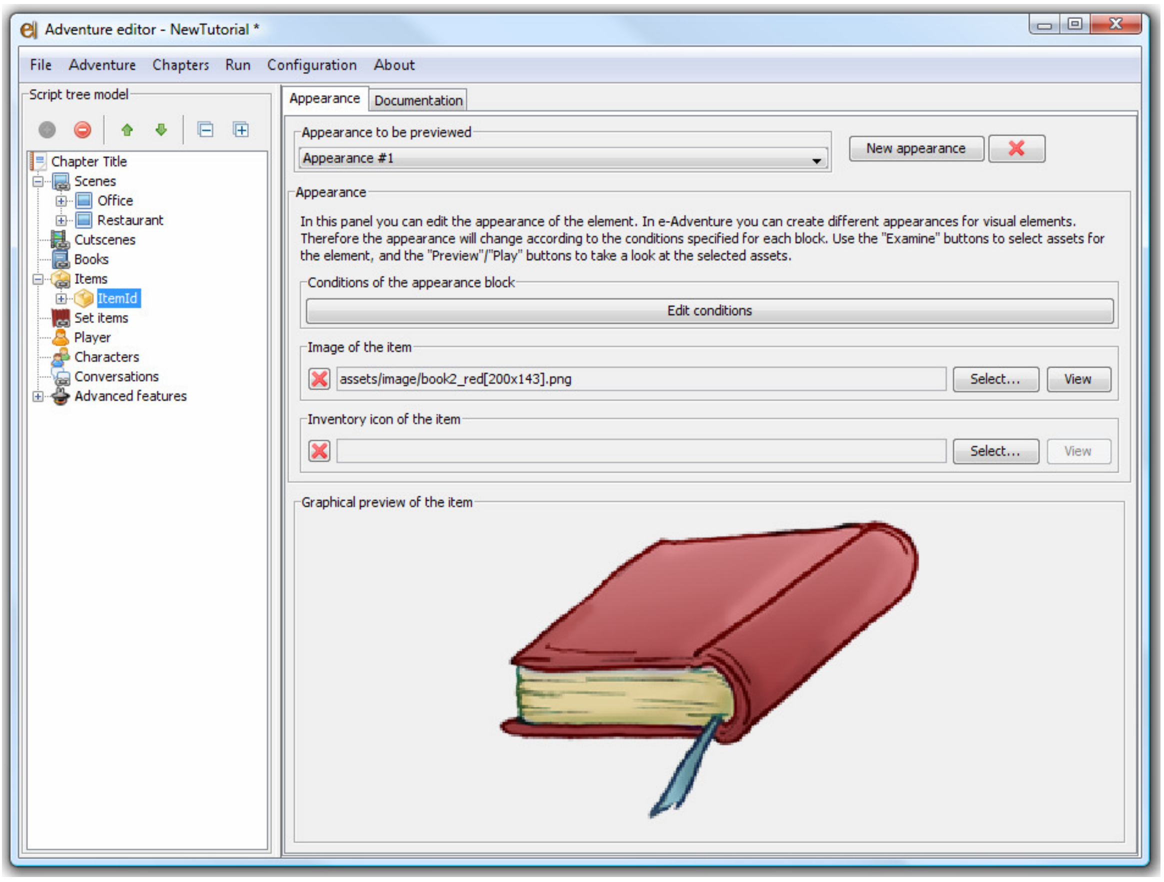


In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

This works as creating new scenes. Here you just need to specify what the item will look like by selecting an image asset, and the icon will be used when the item is in the inventory[1]. Please, remember there are no restrictions for the image, but icons must obligatory to be 80x48.

For instance, we will select an image and icon for our item. We will see its preview at the bottom of the panel:

[1] As in conventional point-and-click adventure games, the player owns an inventory where he puts the objects he collects on his way. Later on he is allowed to interact and somehow give a use to those objects.



In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

assets/image/book2_red[200x143].png

As you see, our item is a book. Hence the id it has been given may not be the most appropriate. We will seize the opportunity to explain how to **rename an item**, which is a simple task. First perform a right-click on the element you want to rename (obviously we are talking again about the left tree). Then press the option *"Rename element"*. You will be prompted to type a new id. In this case we will type "Book", which seems to be more descriptive.

2.3.2. Interacting with items: Actions

Then our basic item has been successfully created. But just give an item an appearance it is not all you can do. If you expand the tree node you will see another node behind it. This one is called "Actions". By clicking on this node you will be able to specify different things the player can do with it. In <e-Adventure> the predefined interactions you can specify for items are:

Examine

Please, do not mistake "examine" with "look". All items in <e-Adventure> can be looked. When you look an item in the game the brief description you type on the documentation tab is displayed, which will be explained later. On the contrary, when an item is examined the detailed description is shown. You cannot change the behaviour of the look action, but the "examine" action is reconfigurable. You can use these two behaviours to give firstly a general scope of what the item is, and later a further description.

Grab

When a "grab" action has been added the player will be able to take the item, which will be removed from the scene and put it in his inventory, the structure where the player stores the grabbed items. Then he will be allowed to use it wherever he is.

Use

Then the item will be usable on its own. That means you could use it and through that use produce somehow effects over the game.

Use with

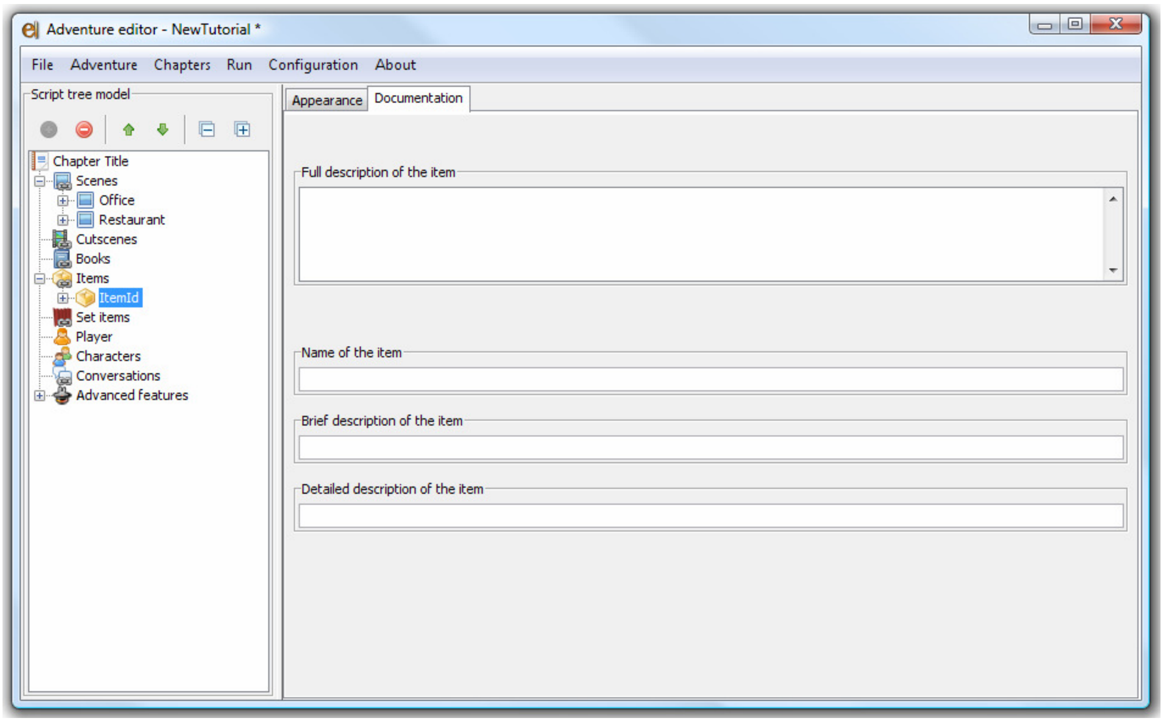
It is equivalent to "use", but in this case the item will not be usable on its own, as you will need to use it with another item to produce those effects associated with the action.

Give to

You can specify the item could be given to a character. The item must be grabbed first, so it is already in the inventory. Then the given item will be consumed and removed from your inventory.

The numbers of actions that can be defined for an item are unlimited. Hence more than one action per type can be defined. Then, how manages the engine to determine which action should be executed in each case? The answer is simple; the game engine will pick the first action of that type which conditions are completely satisfied. Do not worry about this, we will come back to actions when we talk about conditions and effects. This has been just an introduction about what you can and you cannot do with items.

2.3.3. Documentation

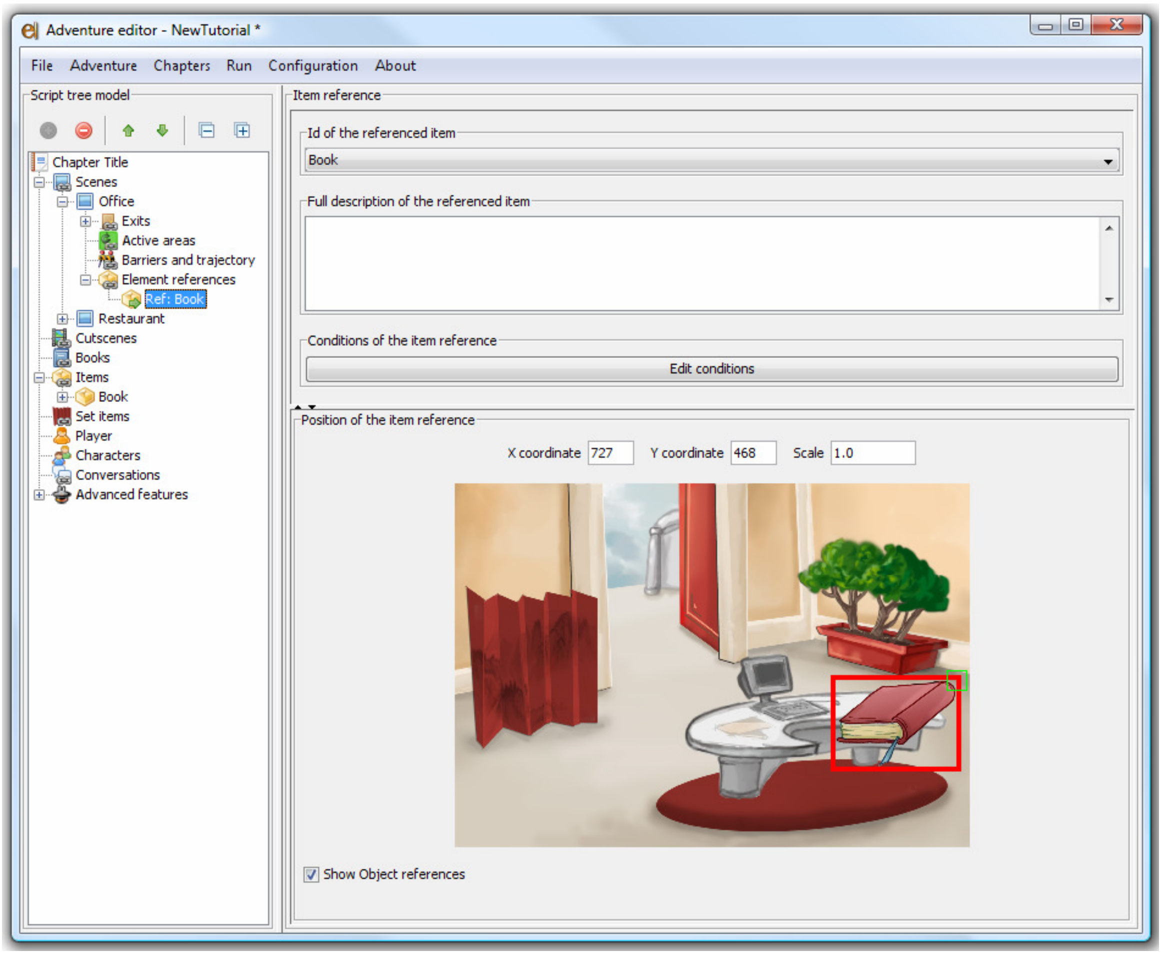


As for scenes, you can attach some documentation to an item. We will have a brief look on this. As you see, there are four fields for you to fill. The first one allows you to specify a complete specification of what the item is. The other three will be used to display info about the item in the game. The name is the piece of text that will be displayed over the cursor when it is on the item. The brief description will be displayed when an item is "looked" and the detailed when it is "examined".

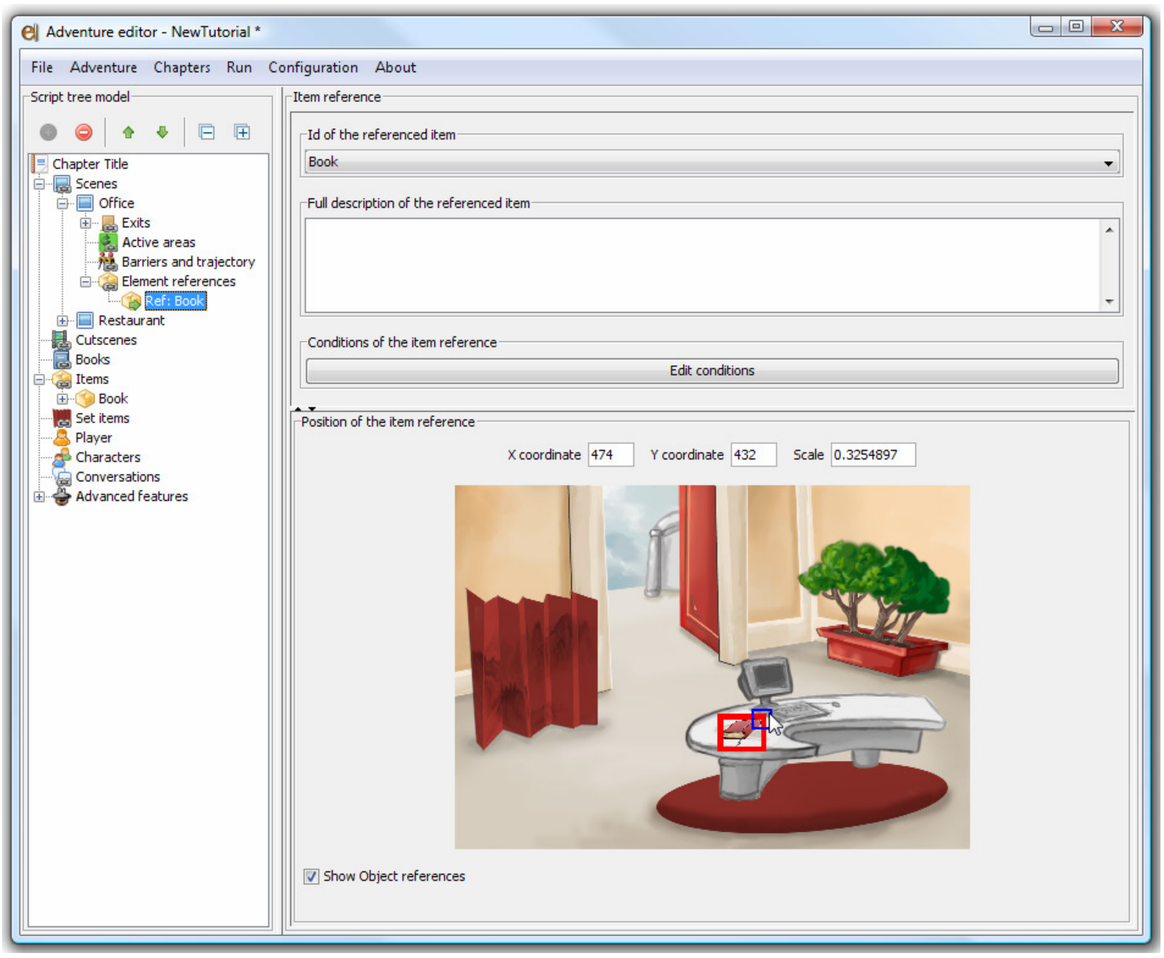
2.3.4. Adding an item in a scene

Once an item has been created, you can refer to it wherever in the chapter. For instance, you can make the item be present in several scenes. Now we will learn how to place those items in the scenes.

The first step is to select the node of the scene in the tree. Then expand its node and you will see a node labeled as **element references**, which is initially empty. After doing right-click on it you can add an item reference to the scene. You will be prompted to select which item you want to refer to, displaying a combo box with all the existing items in the chapter. In our case, we must select "Book" and press ok. Now we just need to click on the preview of the scene to select the position where the item must be:



As you see, the book has been placed on the table. However, the book is too big to fit on the table. You can adjust the scale of the references (not only items but also characters, the player or set items) by dragging the small square which is drawn on one of the corners of the framed item. Now our book scene looks better proportioned:



Id of the referenced item: Book

X coordinate 474 Y coordinate 432 Scale 0.3254897

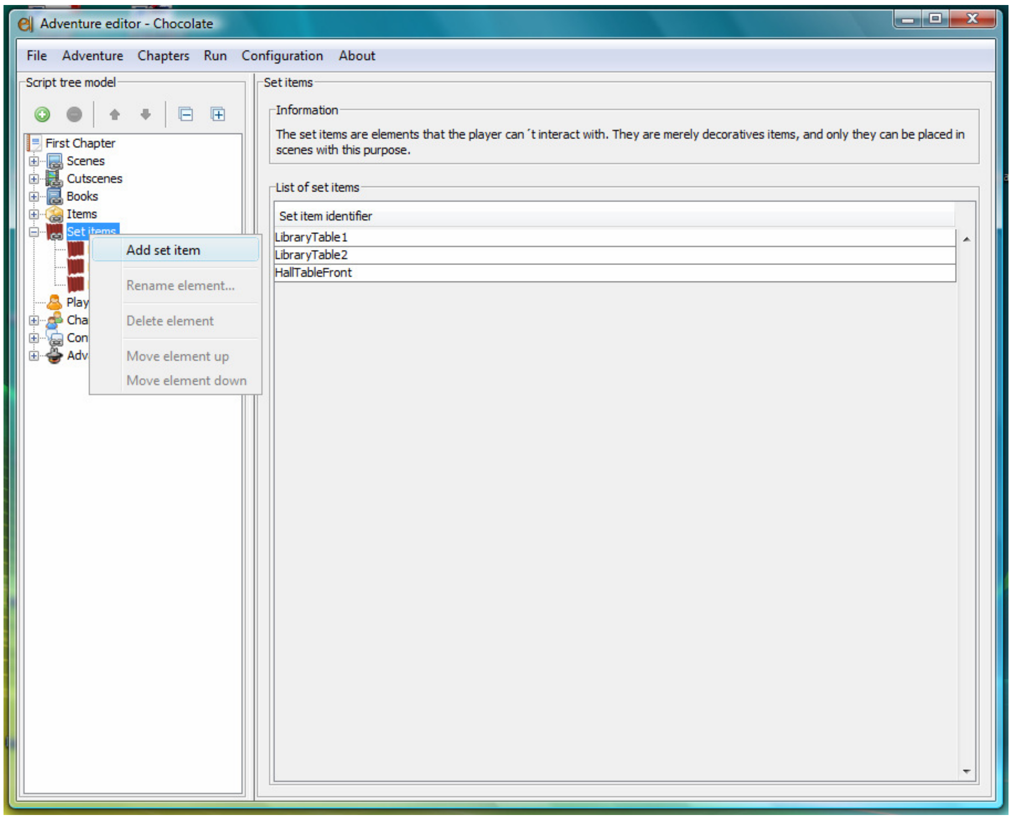
Besides, when you add a reference to an item (or a character) you will be able to configure its "Influence Area". This is just the region of the scene where the player can interact with the object (or character), and is only editable when the scene has trajectories. However, this is an advanced features and thus it is described in the section "Trajectories and barriers" in chapter 3.

2.4. Set items

The set items are merely decorative, as the player cannot interact with them. Let's see how to create such items and how add them in a particular scene.

2.4.1. Create new set item

The set item creation process is similar to create an item. First, you have to do right click on the "***Set items***" tree node. Then a list of all the set items created so far will appear. Later, you have to click on "***Add set item***". Next fill in the set item identifier. It must to be valid and unique.

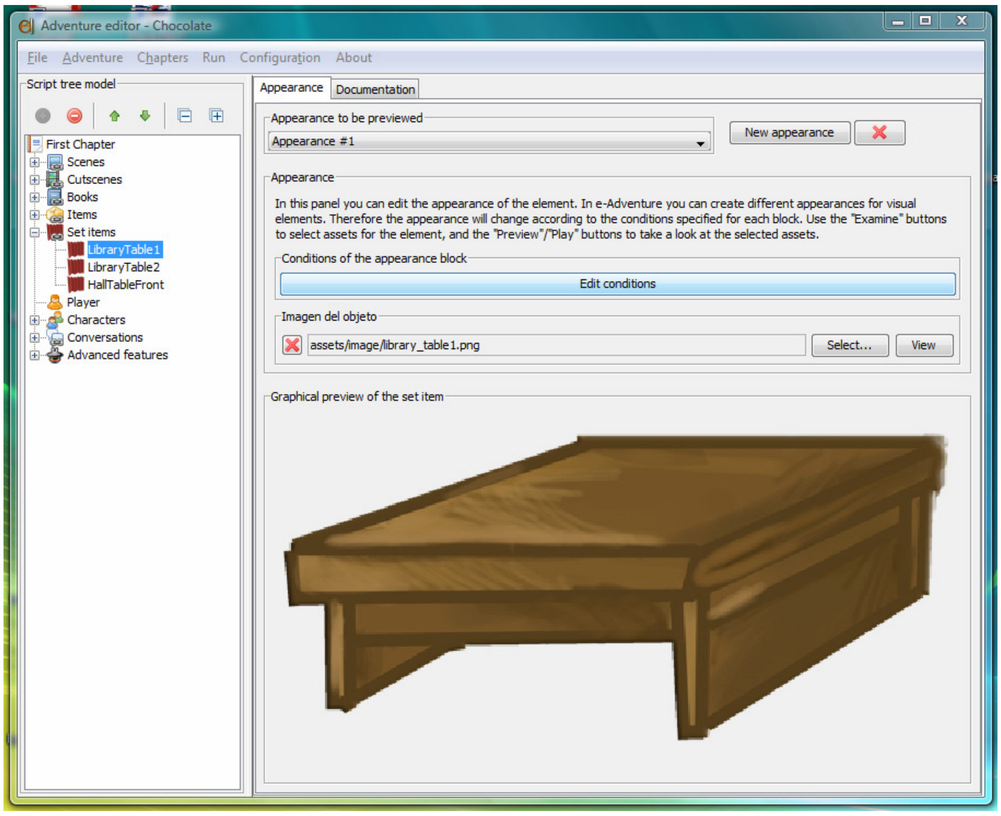


The set items are elements that the player can´t interact with. They are merely decoratives items, and only they can be placed in scenes with this purpose.

Set item identifier
LibraryTable1
LibraryTable2
HallTableFront

Once the set item has been created, we will see the edit panel which is similar to the objects edition panel. First you can found "***Appearance***" tab. This tab lets you assign an image to the set item. There aren't restrictions with image size.

For example, when we choose an image, the following panel appears showing the preview of set item:



In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

assets/image/library_table1.png

As you see, we have added a table. If you introduced a non descriptive id, you can change it in the same way that we described in 2.3.1 section.

2.4.2. Information

It is possible to attach a determinate documentation to a set item. As you can see in main panel in the information tab, there are two editable fields for user information. The first one allows you to give a complete specification of the set item. The second one allows you to give a noun for the set item, but as opposite to objects it will not be used during the game when you pass the mouse over this element.

2.4.3. Add set item in scene

In the same manner with the items, once you have created a set item it can be referred in other parts of the chapter, for example, in the scenes. In this section you will learn to reference set items in scenes.

As in section 2.3, you have to go to left tree and select the corresponding node. In this case, you should select the scene node where you want add set item reference, expand it and look for the *"Element references"* node. The process is equivalent to adding an item reference; the only difference is that you should choose *"Add set item reference"* option (see 2.3.4 section).

2.5. Characters

A character is an element which the player can talk to. Characters (usually known as Non Player Characters – NPC) are like items, but with some differences. On the one hand, a character can be given a set of *animations* so during the game it looks like a living being. On the other, the only interactions you can get from a character is to give them objects and talk to them (characters cannot be grabbed, used or given). In this section you will learn to create and edit all the aspects related to characters.

2.5.1. Creating a new character

The addition of new characters to the chapter follows the same simple rules that items. For that purpose there is a node on the tree labelled as **"Characters"**. Click on it with the right button of the mouse and between other options **"Add NPC[2]"** is displayed. Press on it, type the id for the character and a new character will be created.

The main difference you will realize comparing characters and items is characters allow much more assets to be specified. Those assets are the animations that are to be displayed in different situations. This will give the character the impression that he or she is truly speaking, grabbing something…

When we come to **animations** in <e-Adventure> we must give some details. Firstly, take into account that an animation in <e-Adventure> is a succession of images that will be drawn one after the other. To identify which JPG or PNG images compose the animation, and in which order they must be drawn you must create those images with the same name, but appending the suffix "_index", where index is the two digit number that specifies the order in which the image must be displayed. For instance, an animation called "characterTalking" made up of 5 jpg images will need to get its files named as "characterTalking_01.png", "characterTalking_02.png", "characterTalking_03.png", "characterTalking_04.png" and "characterTalking_05.png", respectively.

[2] NPC is the acronym of "Non Playing Character"

Specifically, the animations you can edit for characters are:

- *Animation for looking up, down and right.*
- *Animation for speaking up, down and right.*
- *Animation for walking up, down and right.*
- *Animation for the use of objects.*

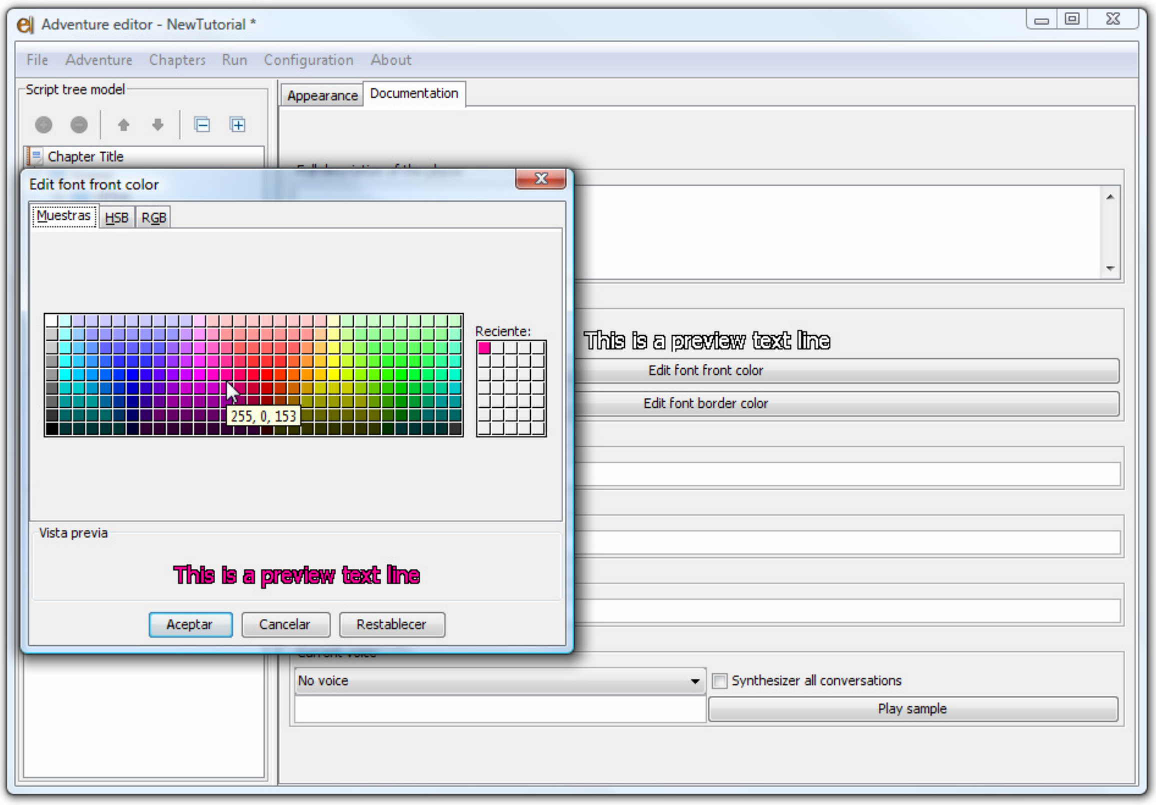
Please, notice that you do not need to specify animations for left directions, as the right animation will be mirrored and displayed.

Besides, since version 0.8 <e-Adventure> includes as well a full animations editor which allows the creation of much more complex animations. To run the animation editor you just need to press the "Create/Edit" button that is associated with every animation. Nonetheless this is an advanced feature and will be described later on in this manual.

2.5.2. Documentation

As well as items, characters may be attached with some documentation. Notice there are two more fields here: the **dialog colour** and the **voice**.

For characters, you can edit what the text said by the character will look like. Specifically, you can edit the colour of both the text and the border. This is very useful when you have conversations where more than one character takes part, so you can differentiate who is speaking thanks to the colour of the line.



Besides the **voice** is a parameter which is used to synthesize the dialog lines of the character as audio. This is useful both for accessibility issues but also to improve the engagement of the conversations without affording the extra cost of hiring actors to record the audio tracks.

In the present release this is a novel feature and is still in the testing phase. Hence the results are note really good. However all the voices can be tried out using the editor and their use is deactivated by default.

In fact there are ways to synthesize conversation lines. One is to select "***Synthesize all conversations***" field of the panel above, along with the voice of the character. The other option is to activate the voice synthesis line by line in the conversations. This allows you to select which lines you want to synthesize. You can find more information about it in the conversations section.

2.5.3. Add characters in scenes

Exactly as for items, you can make references to characters in the scenes. Then the character will be present in the scene, so the player could interact with him / her. To achieve this you only need to go to the scene, press with the right button on the node **"Element references"** and click on add character reference. Then select the character and place it on the scene.

2.5.4. Conversations

As you can interact with items, you can do with characters as well. But this interaction is significantly different. For characters you cannot specify actions. In contrast you can specify "conversations", which are successions of text lines that are spoken by either the main player or a character. Those are very useful for two purposes: to guide the player in the game, and to provide sometimes a source of information or evaluate the student through a multiple choice test.

More than one character and the player can take part in a conversation. As in conventional *Lucas Arts*TM games the player can be prompted sometimes to select what line must be said next from a list of options. Depending on the option selected the conversation will take one path or another.

Conversations are attached to characters, so they are triggered when the player comes next to that character and selects to talk to him/her. The process of creating a new conversation and attach it to a certain character, you have to use the "conversations" node on the tree. Once the conversation is created, you can attach it to a character.

Now we will see how to create conversations. Firstly you must know there are two types of conversations.

Tree conversations

A tree conversation is suitable when no loops must be supported in the conversation. That is the conversation always reaches an end. To create a new tree conversation, press with the right button on the "conversations" node of the tree, type the id of the conversation and you will be ready to edit it. The conversation will be drawn as a set of nodes (painted as black circles) which are linked to other nodes. There are **three types of nodes**:

- *Dialog nodes*

Those nodes contain a set of lines which are spoken by the player or a character, in the order specified.

- *Option nodes*

Those nodes contain some options to choose one when the conversation reaches that node. The option selected will be the next line spoken by the player, and it will move the conversation from this option node to other node.

- *Terminal nodes*

Those are dialog nodes. The only difference is that those nodes cannot be linked to any other, as the conversation will end when they are reached (after the dialog they contain has been completely spoken).

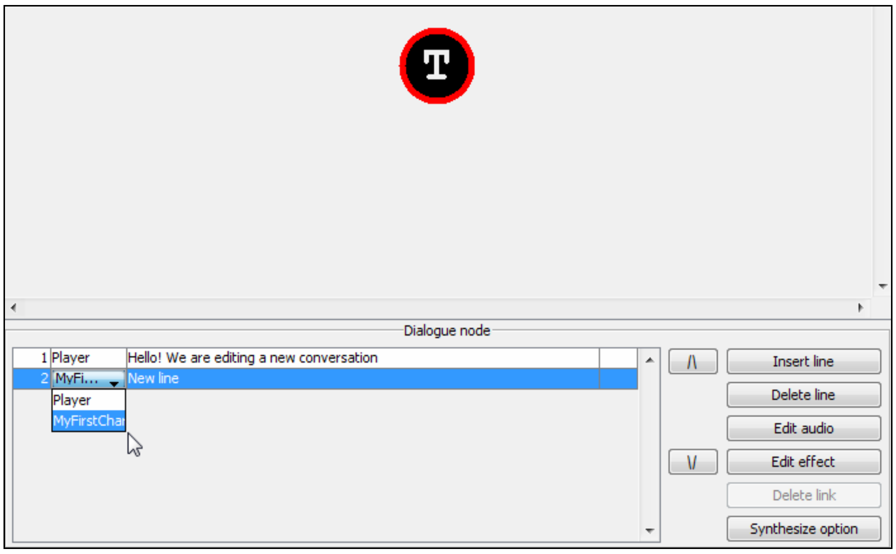
Let's start editing the conversation. You can select whatever node you have already created by left-clicking on it. Then its outline will be drawn red, instead of black, and the fields you can edit will be displayed on the bottom. With the right button of the mouse you can display options to link in different ways this node to others.

For instance, we will select the single node you have when the conversation is created. You will notice it is terminal as a big "T" is drawn inside. Then we use the **"Insert line"**, and **"Delete line"** buttons to insert and delete new lines to be spoken by the player or other characters. The up (/\) and down (\/) buttons can be used to change the order in which conversation lines must be said. When a line is selected on the table, use these two buttons to move it up or down, being therefore said in a previous or later position. You can also specify an audio track to be played along with the text of the line. For that purpose use the button **"Edit audio"**. Thanks to that you can record the text of the lines in audio files, and then get them really spoken in the game.

There are three more buttons you can use. The first one, labeled as **"Edit effects"** will allow you to specify a block of effects to be executed after all the lines of the node had been spoken. The second, **"Delete link"**, can be used to delete links between nodes. Finally **"Synthesize option"** allows activating the automatic synthesis of the dialog line.

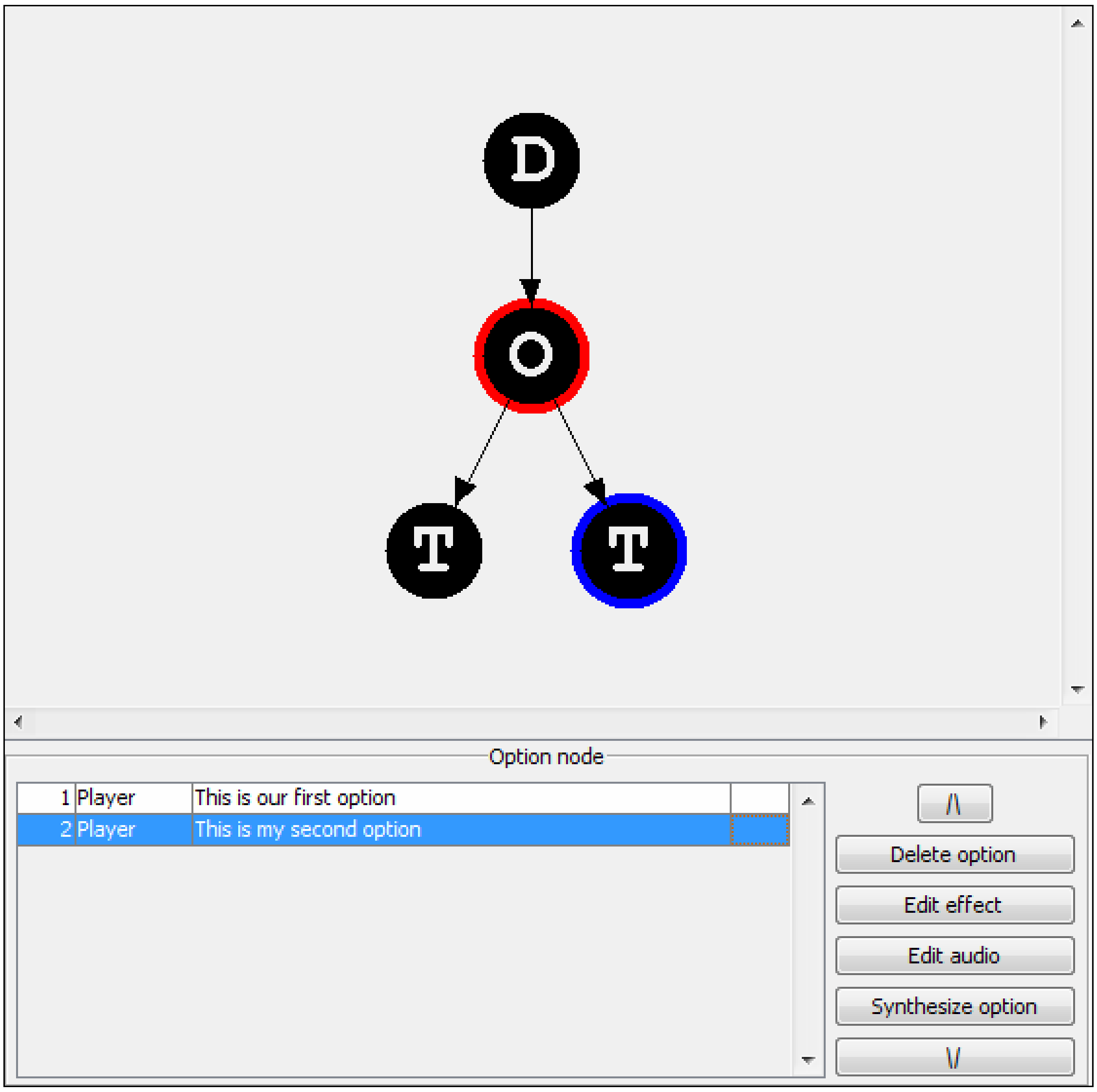
To synthesize conversation lines previously you should have chosen a player or character voice (see section 2.5.2). When you use "***Synthesize option***" button, a dialogue appears showing the player or character chosen voice which will reproduce the line. Also, in this dialogue you can see "***Play sample***" button, which allow reproduce how the current line will be synthesize. If you want in-game synthesizing, you ought to select "***Synthesize current line***".

Now you know all you need, let's edit the conversation. For instance, we can insert two lines. The first will be spoken by the player and the other by our character. As you see, for each line you can change who will say it through a combo box and type the text of the line.



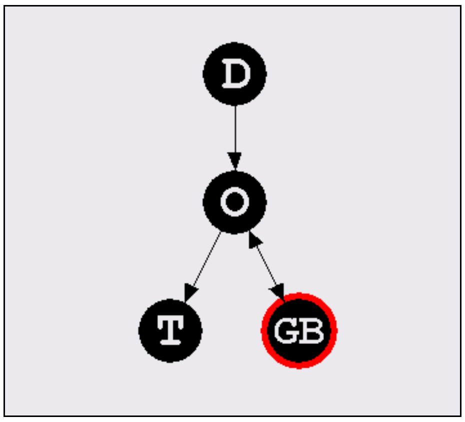
1	Player	Hello! We are editing a new conversation
2	MyFi...	New line

Now, we want the player to choose between two options. Then an "options" node must be created and linked to the current node. To do that, click with the right button on the node and select "Add option node". Once you have done that you will see a new node, with the text "O" drawn inside, hanging from the first node. This now is not a terminal node any more, as there are other nodes below it (hence a "D" is drawn inside instead of the initial "T"). To add options in this node select it, do right-click and select **"Add options node"**. You will see in the panel below a new line has been created along with a new dialogue hanging to the options node. If you select that option line, the node to which that option leads will be blue-outlined. Now we type the text of the option (which will be drawn in the list of options and said by the player when selected). For instance we type "This is your first option". Now we create another option by selecting the options node and adding a new dialogue node. For that option we type "This is your second option". At this point our conversation will look like this:

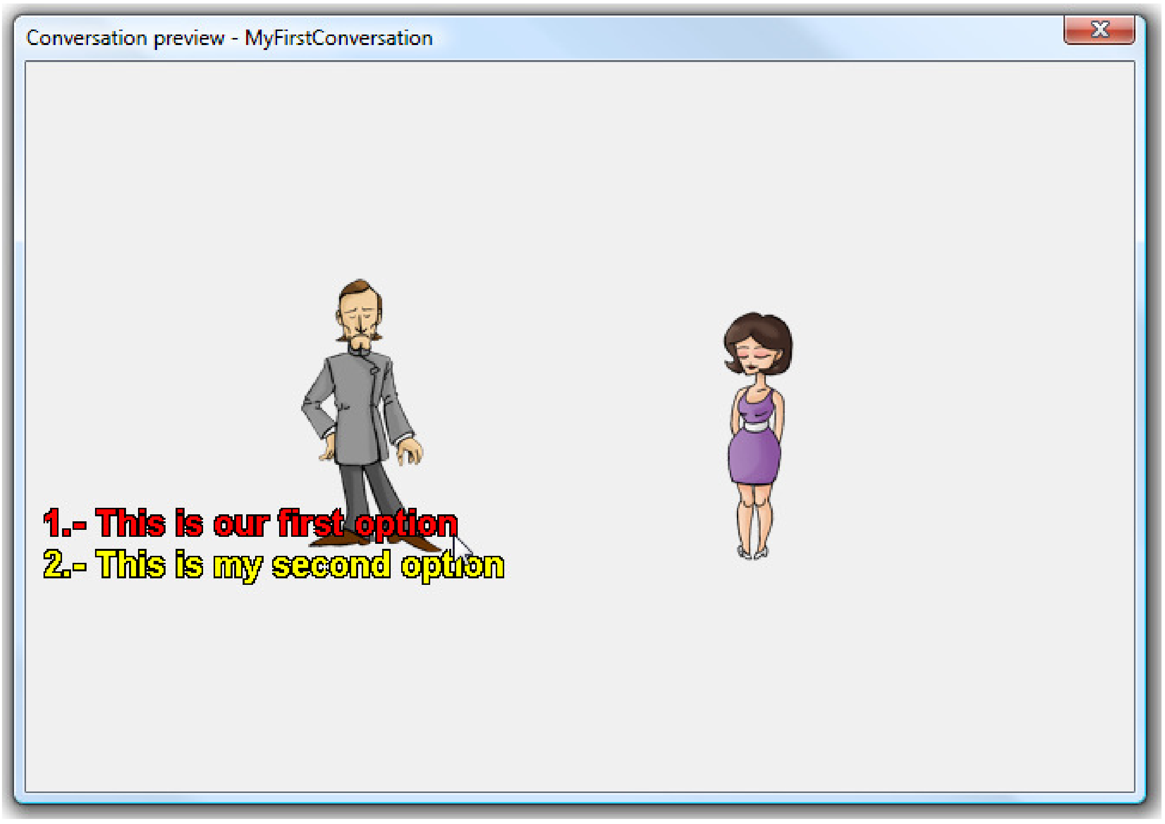


1	Player	This is our first option
2	Player	This is my second option

Following this mechanism you can create tree-conversations as big as you want. However, sometimes you would like to add options in an option node to come back to that node. In this manner, when that option is selected All the options will be displayed and the player will need to choose another one again. That can be achieved by right-clicking on the option we want to make the conversation move back, and select then the option **"Add go-back tag"**. Then, when all the lines in that node had been said, the conversation will return to the options node.



When we had finished editing the conversation, we can preview how it will look in the game. To do that press the right button of your mouse on any node, and select the option **"Preview conversation"** or **"Preview conversation from this node"** if you just want to preview a piece of it.



1.- This is our first option
2.- This is my second option

In addition, the options of an options node can be prompted in a random order to the player. To get this you only need to activate the option **"Order options randomly"** on the pop-up menu that appears after a right click on the node. This is especially adequate when a conversation is used as a multiple-choice test to evaluate the students, as it makes cheating harder.

Graph conversations

In addition to tree conversations, you can create graph conversations in a very similar way. The difference with tree conversations is loops are permitted here. Hence you can link a node to whatever other node you want. To make this process easier, you can move the nodes along the panel, so they will not overlap each other.

Attaching a conversation to a character

Now the conversation has been created, you must attach it to a character if you want it to be triggered by interacting with that character. This is not the only way to trigger a conversation, as there is an effect for that purpose. If you wish to do that, select the character on the tree. You will see there is a node hanging from the character node, labeled as "Conversation references". Use this node to add a new conversation reference. Then you only need to select which conversation you want to refer to.

2.5.5. The player

The player represents the avatar handled by who is playing the game. Is this character who interacts with other characters or items. The fields you can edit here are exactly the same as for characters. The only difference is a player cannot own any conversation.

There are two modes for the player: *Visible* (adventures in *Third Person* mode) and *Transparent* (adventures in *First Person* mode). To learn about that see the last section of this document.

2.6. Books

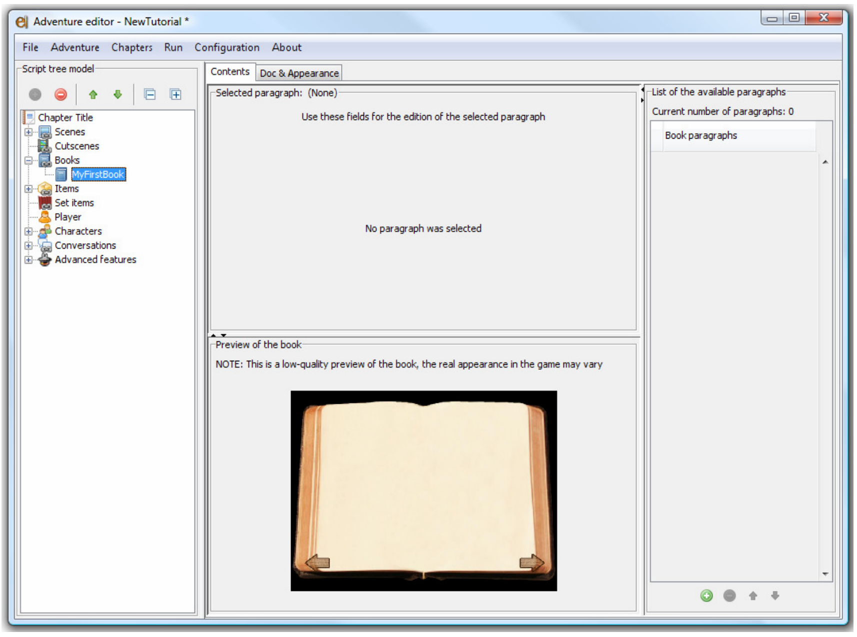
Conversations are a good source of information, as it has been exposed previously. Nonetheless, <e-Adventure> is devised to produce educational videogames, which will require sometimes the delivery of huge amounts of data to the learner. For that purpose you can include in-game books in <e-Adventure> games. Those in-games will be available when and how you specify, and can be produced in two different ways.

Firstly, you can produce **simple books** without the support of any other authoring tool, just using the adventure editor. This kind of books may contain a succession of text, bullet and image paragraphs. The engine will deal with those paragraphs, placing them one after the other in the book, and using different pages if they do not fit in one page. The books produced of this type will be simple, but easy to create as you do not need to deal with pagination, margins, etc.

The second type of books is known as **formatted text books**. Those books will have a richer appearance, but to produce them you will need an external authoring tool. You can use Microsoft Word, Open Office, FrontPage or any other tool that supports the creation of RTF (Rich text format) or HTML documents. Those documents will be used as pages, so then the pagination is not managed by the engine.

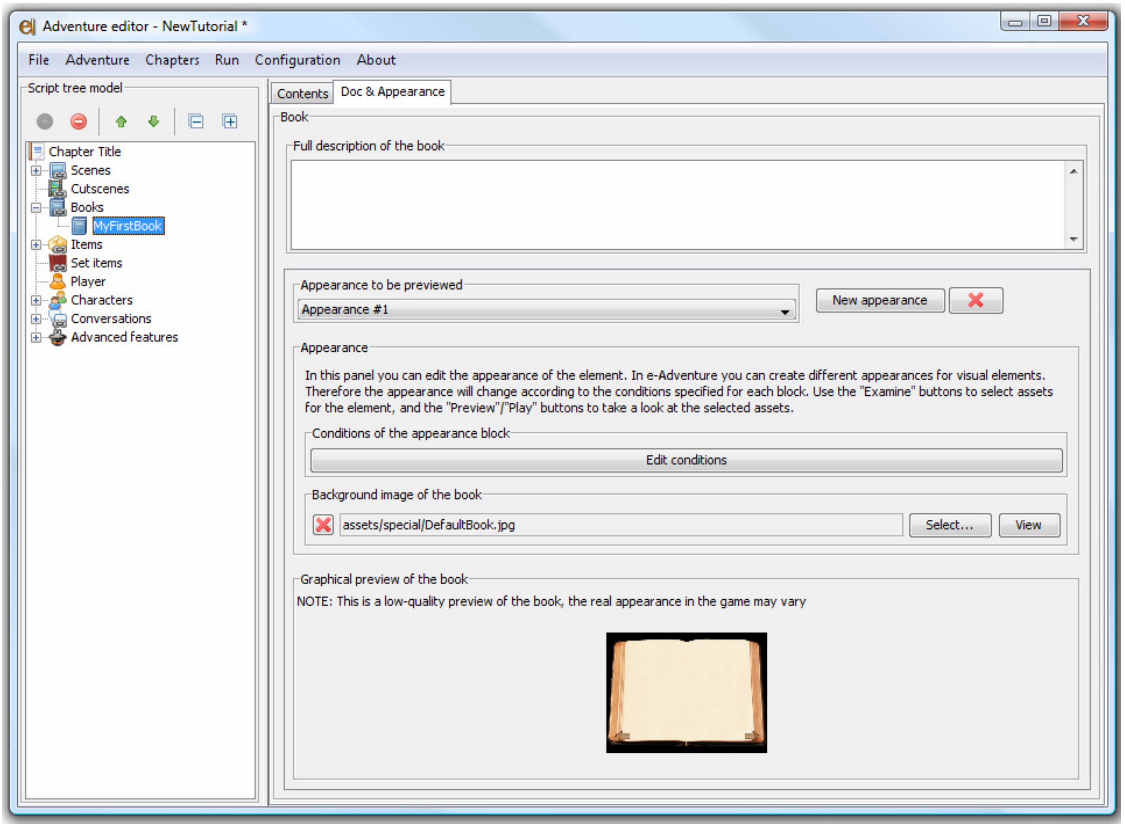
However, the present <e-Adventure> release has embedded a simple HTML editor. Then you can use it to modify and create your HTML documents, as section 3 describes.

Let's give a better explanation using an example. Go to the left panel, find the "Books" node and add a new book following the same process for other elements. Then you will be prompted to select between two types: "Simple content books" and "Formatted text books". Select the first option and you will see a panel like this:


Adventure editor - NewTutorial *
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
Cutscenes
Books
MyFirstBook
Items
Set items
Player
Characters
Conversations
Advanced features
Contents
Doc & Appearance
Selected paragraph: (None)
Use these fields for the edition of the selected paragraph
No paragraph was selected
Preview of the book
NOTE: This is a low-quality preview of the book, the real appearance in the game may vary
List of the available paragraphs
Current number of paragraphs: 0
Book paragraphs

As you will notice, two tabs are available here. The *Contents* tab supports the creation, edition and organization of the paragraphs of the simple book. The second tab, labeled as "Doc & Appearance" is very similar to the other panels you have seen so far. There is an asset that you can edit, which is the background image which will be used to give the appearance of a real book. By default the adventure editor loads a background image for the books, so you do not need to customize this if are happy with it. However, if you decide to use your own background keep in mind that the arrows at the bottom are the places where the user must click to change the page. If the new image does not show those places, the user might not be able to browse the current page.

In our case, we will leave the default image:


Adventure editor - NewTutorial *
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
Cutscenes
Books
MyFirstBook
Items
Set items
Player
Characters
Conversations
Advanced features
Contents
Doc & Appearance
Book
Full description of the book
Appearance to be previewed
Appearance #1
New appearance
Appearance
Conditions of the appearance block
Edit conditions
Background image of the book
Select...
View
Graphical preview of the book

In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

assets/special/DefaultBook.jpg

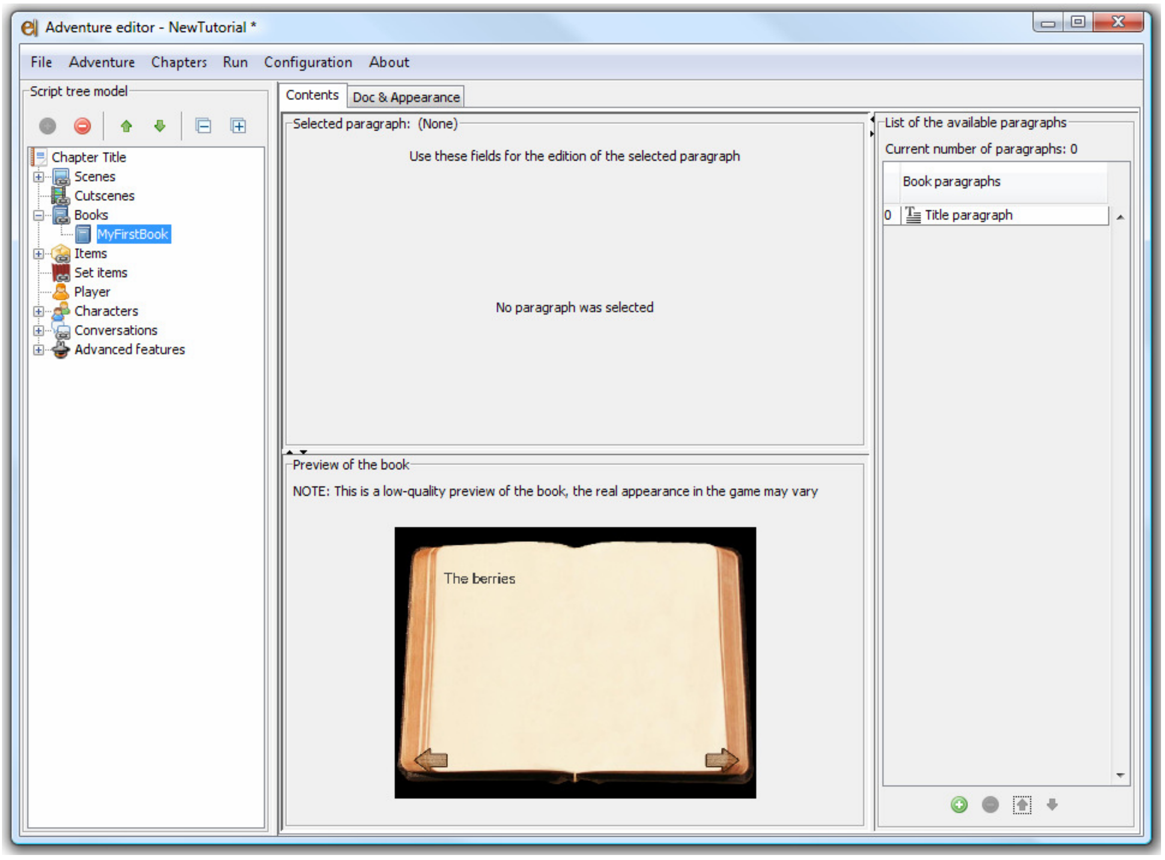
NOTE: This is a low-quality preview of the book, the real appearance in the game may vary

2.6.1. Creating new paragraphs

Then we can start creating paragraphs. You can add as many paragraphs as you desire, and the engine will automatically spread them in several pages when they do not fit in a single one. Use the "Contents" tab to manage the paragraphs of the book.

For instance, we are going to create a book with a **text paragraph** and an **image paragraph**. We will also create a **title paragraph**, which is the same as a text paragraph but which will be rendered using a larger font size to make it look "more important". There is another type of paragraph supported by <e-Adventure>: a **bullet paragraph**. This will be useful when you want to organize text in several points. We will use our book to give a brief explanation of what berries are.

Let's start adding a new title paragraph. For instance, we will type as a title the text "This is the title of the book".



Selected paragraph: (None)

Use these fields for the edition of the selected paragraph

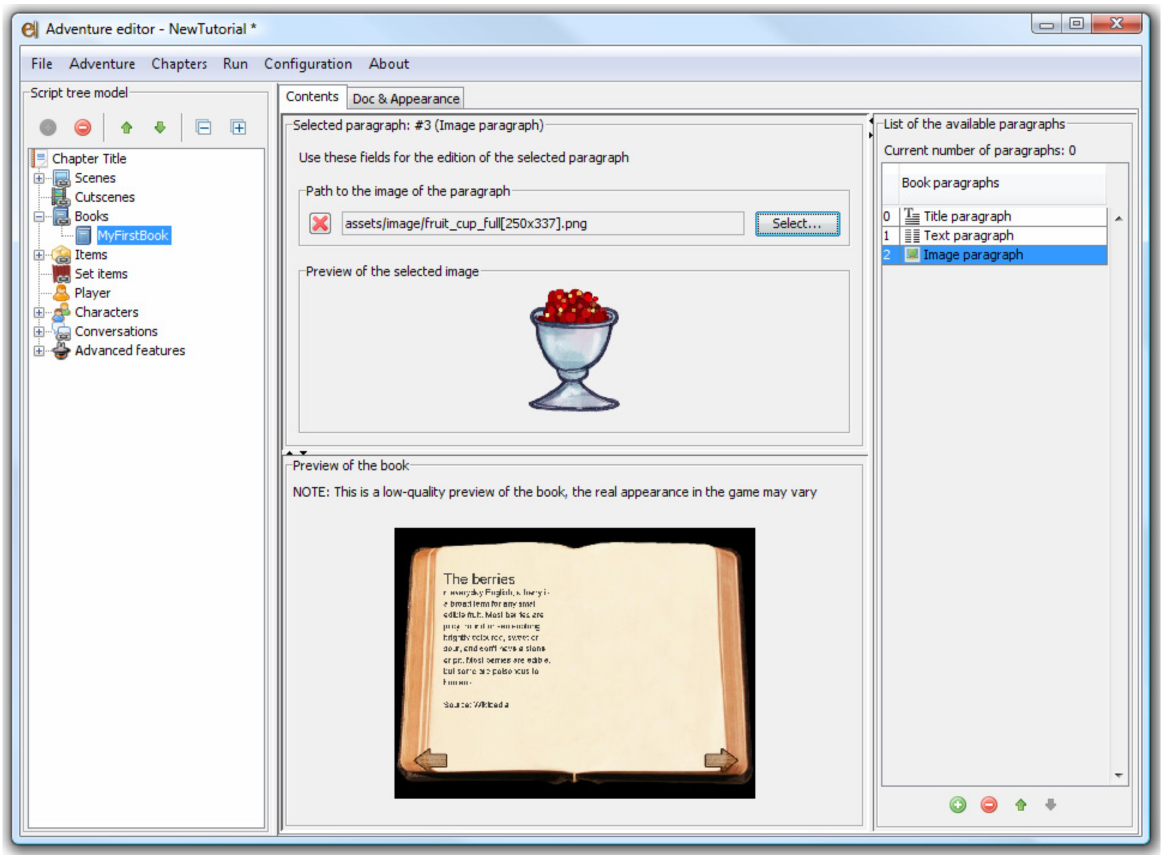
No paragraph was selected

NOTE: This is a low-quality preview of the book, the real appearance in the game may vary

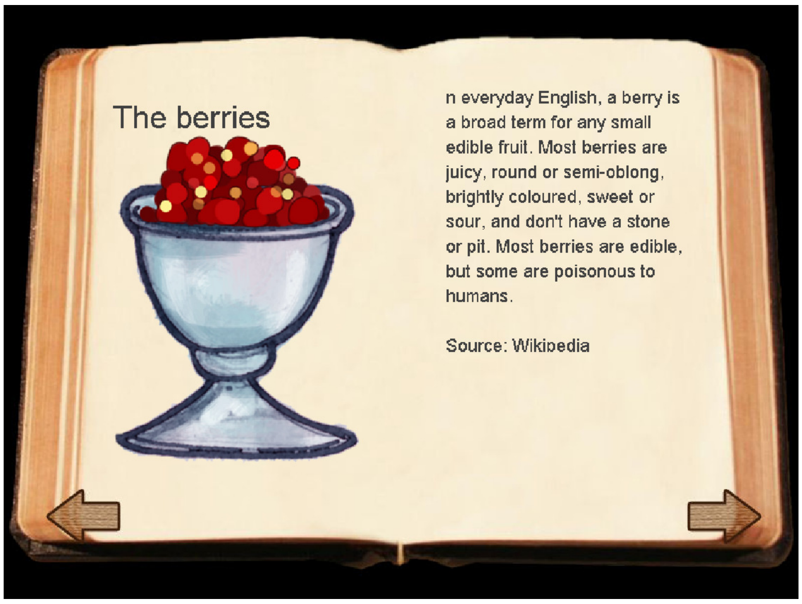
Current number of paragraphs: 0

The second step is to create a Text paragraph. Here we type the text we want to include, for instance an extract obtained from Wikipedia about berries.

At last, we add the image the image paragraph. Here we can specify which image we want to be drawn. In our case, we select the image of a cup of berries.


Adventure editor - NewTutorial *
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
Cutscenes
Books
MyFirstBook
Items
Set items
Player
Characters
Conversations
Advanced features
Contents
Doc & Appearance
Selected paragraph: #3 (Image paragraph)
Use these fields for the edition of the selected paragraph
Path to the image of the paragraph
assets/image/fruit_cup_full[250x337].png
Select...
Preview of the selected image
Preview of the book
NOTE: This is a low-quality preview of the book, the real appearance in the game may vary
List of the available paragraphs
Current number of paragraphs: 0
Book paragraphs
0 Title paragraph
1 Text paragraph
2 Image paragraph

Finally, if we click on the book node we will see the preview of the book:


The berries
n everyday English, a berry is a broad term for any small edible fruit. Most berries are juicy, round or semi-oblong, brightly coloured, sweet or sour, and don't have a stone or pit. Most berries are edible, but some are poisonous to humans.
Source: Wikipedia

2.6.2. Formatted text books

Now we are going to re-implement the cherries book using HTML to achieve a better looking.

We must give some initial considerations regarding the creation of HTML content for <e-Adventure> books with external tools. This is an experimental functionality and hence how the HTML documents are processed and displayed in <e-Adventure> might have an unexpected behaviour.

So far we have detected that HTML documents containing meta-data ("meta" labels within the "head" label) do not work properly in <e-Adventure>. Therefore, if for instance those documents are edited using Microsoft Word the files must be saved as filtered web pages and after that the meta-data must be manually removed. Next versions of <e-Adventure> this functionality will be reviewed and the HTML documents will be adapted automatically.

Another important point is how the files which are attached to an HTML document (e.g. the images embedded in the file) must be placed and named so the <e-Adventure> editor could know how to retrieve them and copy them into the adventure project. The convention adopted in <e-Adventure> is that those files must be located in a folder named as the HTML document but with the suffix "_files" at the end. For instance, for the document "cherries.htm" the editor will copy the contents of the folder "cherries_files", if it exists.

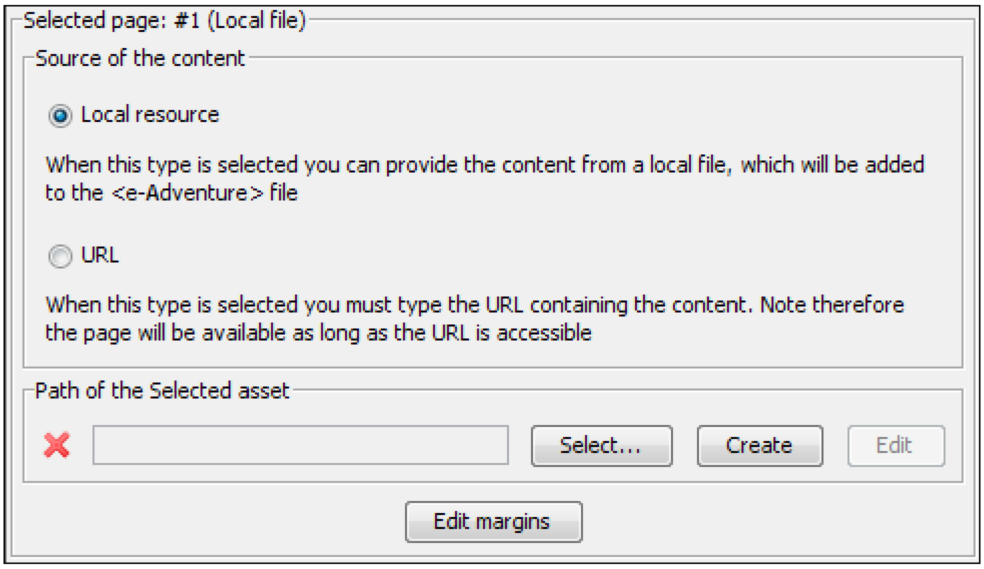
Nonetheless, you can also use the HTML editor that is included in <e-Adventure> since version 0.8, as it is described in the third section of the document. In such case there is no need to take into account those considerations.

Once it is clear how to produce valid HTML documents for <e-Adventure>, let's start. Suppose we have re-implemented the pages described in section 2.6.1. using an HTML editor, such as FrontPage, obtaining the next document:

Las grosellas

La grosella es ligeramente más ácida que su pariente la grosella negra, y se cultiva principalmente para producir mermeladas y platos cocinados, en lugar de consumirse fresca. En Escandinavia se suele emplear en la elaboración de sopas y postres de verano, y en Alemania y en los Países Bajos se usa en diversos rellenos para tartas.

The text presents different fonts and colour settings. Now we are going to create a book using this document. The first step is to add a new book (see section 2.6.1) and select "Formatted text book". The edition panel of the new book looks very similar to the one described in 2.6.1. Now the right panel supports the basic administration of the pages of the book (that is, add and delete pages, alter their order, etc.). After adding a new one the fields of the page that can be edited are displayed on the left panel.



Selected page: #1 (Local file)

Source of the content

Local resource [✓]

When this type is selected you can provide the content from a local file, which will be added to the <e-Adventure> file

URL

When this type is selected you must type the URL containing the content. Note therefore the page will be available as long as the URL is accessible

Path of the Selected asset

As the figure above depicts, we can choose between two different sources to retrieve the HTML document from. The first option is to include the HTML document as local resource, supposing that the file is in our system. If we select this option (which is the default option) the selected file will be automatically copied to the project folder.

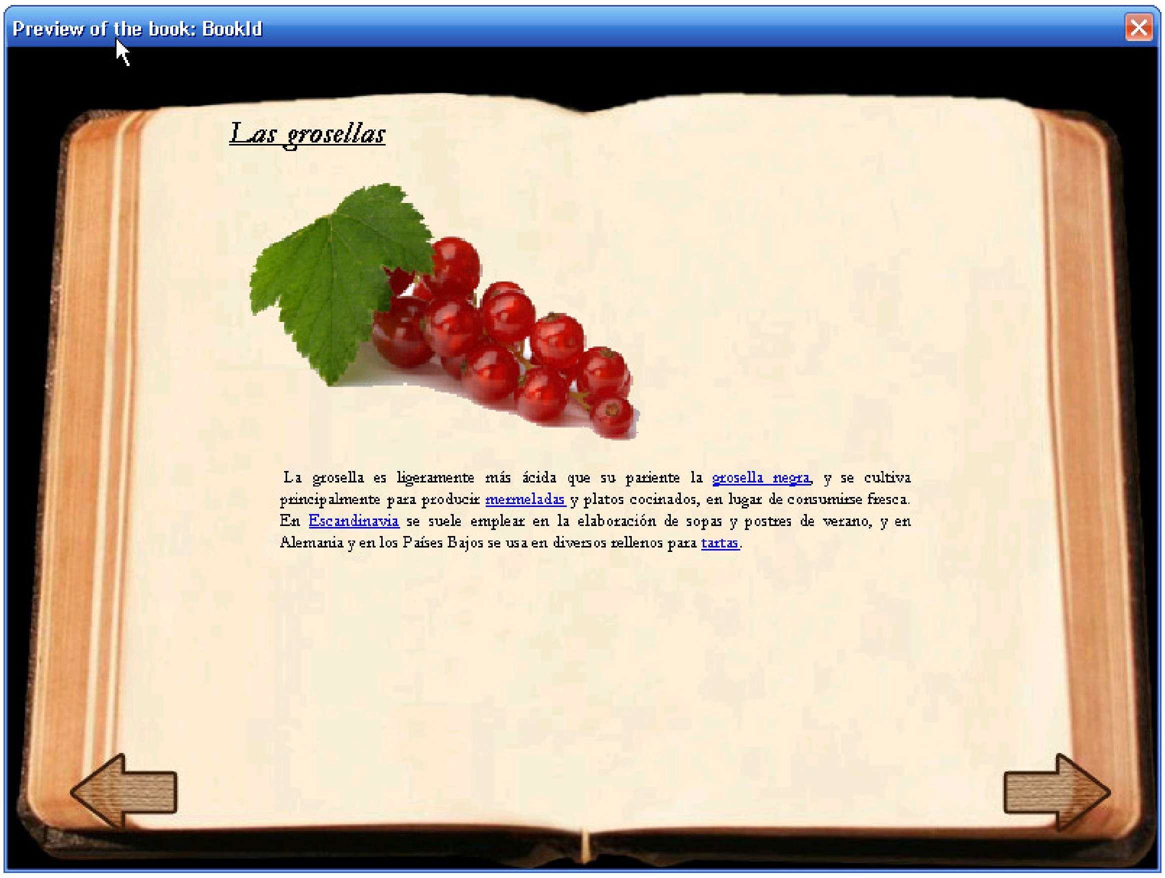
On the other hand we can provide the editor the URL of the document. The advantage of this approach is that the document does not have to be compulsory in the system, as it can be placed wherever on the Internet. Hence, if the document changes, there is no need to update the game as it is retrieved online every time the game is launched.

Under the radio buttons we find the utilities to select the path/URL of the document, create a new one or edit the existing one (only for local resources). The left icon will turn into a green tick if the document is correct and a red cross otherwise. This is especially useful when the source of the document is a URL.

Finally, at the bottom there is a button, **"Edit margins"**, which is useful in order to adjust the layout of the page. If the button is pressed a dialog with a full-size preview of the book will be shown on the screen. This dialog contains 4 slide bars which can be used to gauge the top, bottom, left and right margins (the margin is the distance between the edge of the screen and the beginning of the document).

All the changes on the book will be updated on the small preview that can be found at the bottom of the window. However, to get a full screen preview there is a special facility at the Information tab.

This is how the book looks like after selecting the document from the file system and adjusting the margins.



Las grosellas

La grosella es ligeramente más ácida que su pariente la grosella negra, y se cultiva principalmente para producir mermeladas y platos cocinados, en lugar de consumirse fresca. En Escandinavia se suele emplear en la elaboración de sopas y postres de verano, y en Alemania y en los Países Bajos se usa en diversos rellenos para tartas.

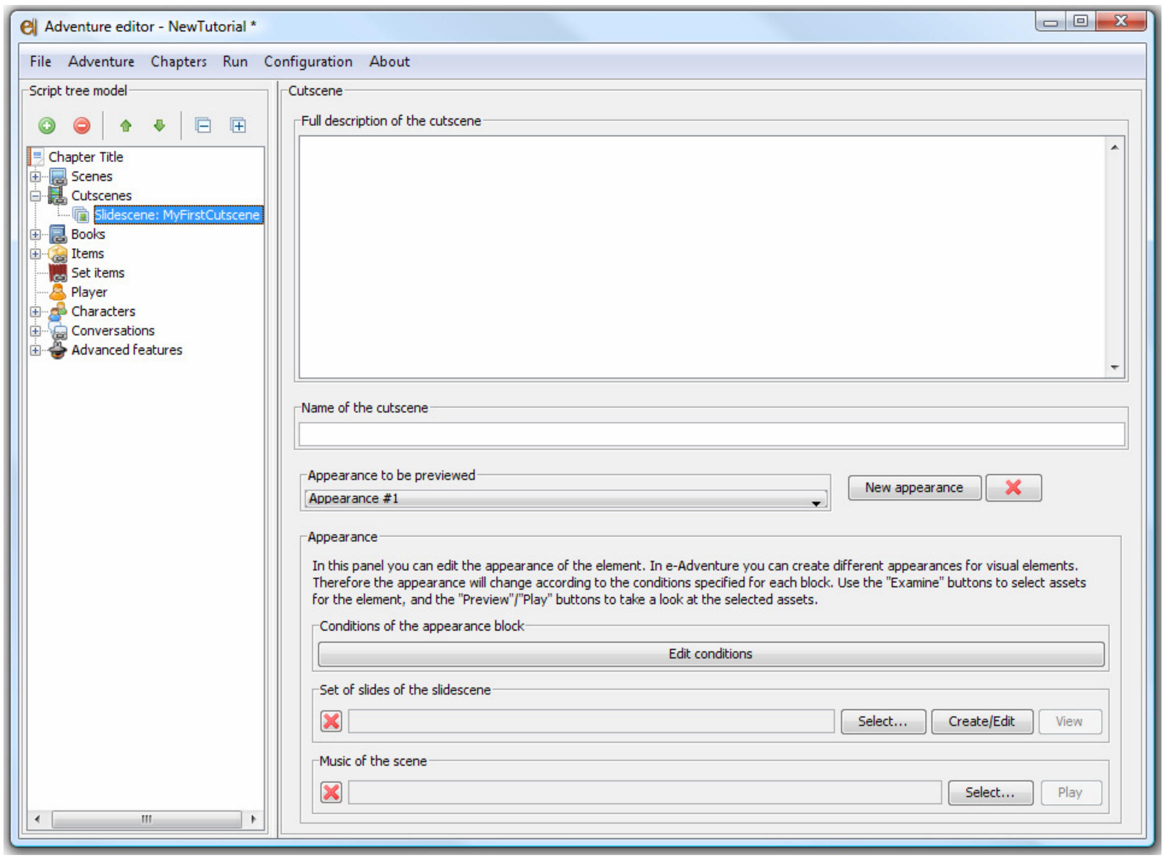
2.7. Cut-scenes

As you have seen so far, <e-Adventure> games are structured and organized in scenes where the action takes place. However, you can use other types of scenes, which differ completely from the concept of a scenario where items and characters are placed. These are **cut-scenes**, pieces of media documents which are full-screen displayed when the player reach them, and which lead to other scenes or cut-scenes when they are done.

These cut-scenes are of two kinds: slide-scenes and video-scenes. As you can imagine, the first is a set of images which are displayed one after the other (as slides), and the other is just a video in MPG, MOV or AVI format.

2.7.1. Slide-scenes

Let's begin creating a new slide-scene. This is as easy as creating new elements have been so far. Hence, perform a right-click on the node **"Cutscenes"** and select **"Add slidescene"**. Select a valid id for the slide-scene and then you will get a panel on your right like the one below:

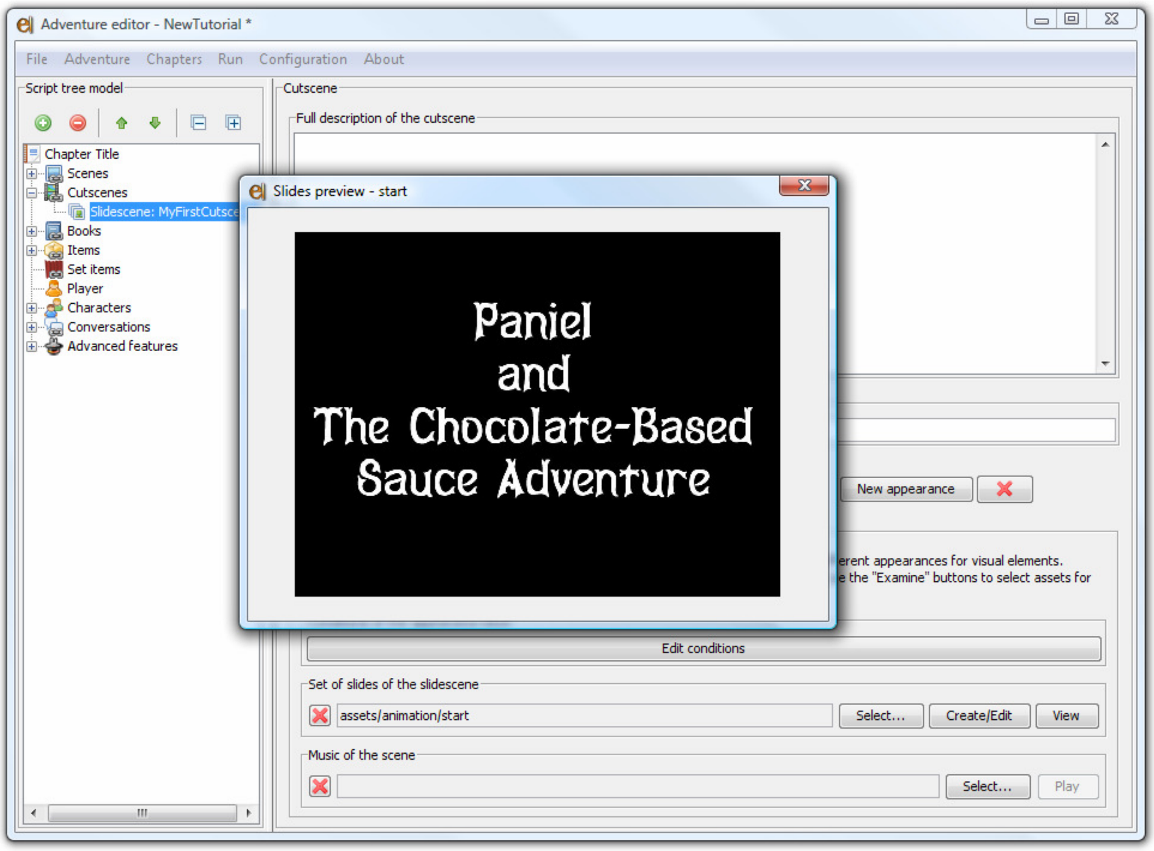


In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

The first allows you to set a description of the cut-scene. There you can specify what is for, what is it about, educational goals, etc (i.e. you are free to attach the information you think is relevant). The second is just a text field where you can type the name of the scene. This name will be used when the player locates the mouse on an exit which leads to this cut-scene. However, both areas are optional, you do not compulsory need to fill them.

The core which you are really obliged to select is the set of slides resource embedded in the "Appearance" panel. Use the select button to deploy a chooser, similar to the others you have previously used. Then choose a valid slide-scene. Remember that all the slides of a slide-scene, as it happened with animations, must have the same file name but appending the suffix "_dd" at the end. For instance, a set of slides named "slides" made up of 4 slides will require naming the images "slides_01.ext", "slides_02.ext", "slides_03.ext" and "slides_04.ext" respectively, where ext can be jpg or png (Note: all the images must have the same format). In addition, the <e-Adventure> editor has an advanced edition mode for animations that you can use, as section 3 describes.

Slide-scenes are very useful as titles as well. You can start and end all your chapters with a single-image slide-scene with the title of the chapter and the text "The end" respectively. The result would be something like this:


Adventure editor - NewTutorial *
File Adventure Chapters Run Configuration About
Script tree model
Chapter Title
Scenes
Cutscenes
Slidescene: MyFirstCutsce
Books
Items
Set items
Player
Characters
Conversations
Advanced features
Cutscene
Full description of the cutscene
Slides preview - start
Paniel
and
The Chocolate-Based
Sauce Adventure
New appearance
erent appearances for visual elements.
e the "Examine" buttons to select assets for
Edit conditions
Set of slides of the slidescene
assets/animation/start
Select... Create/Edit View
Music of the scene
Select... Play

2.7.2. Video-scenes

Creating and editing a new video-scene is as simple as for slide-scenes. Just press in the node **"Cutscenes"** and select **"Add videoscene"**. The panel is the same you got for slide-scenes. Now, you just need to select your video by pressing the **"Select…"** button. You can use either the chooser to preview the video or the **"Play"** button you will find next to "Select".

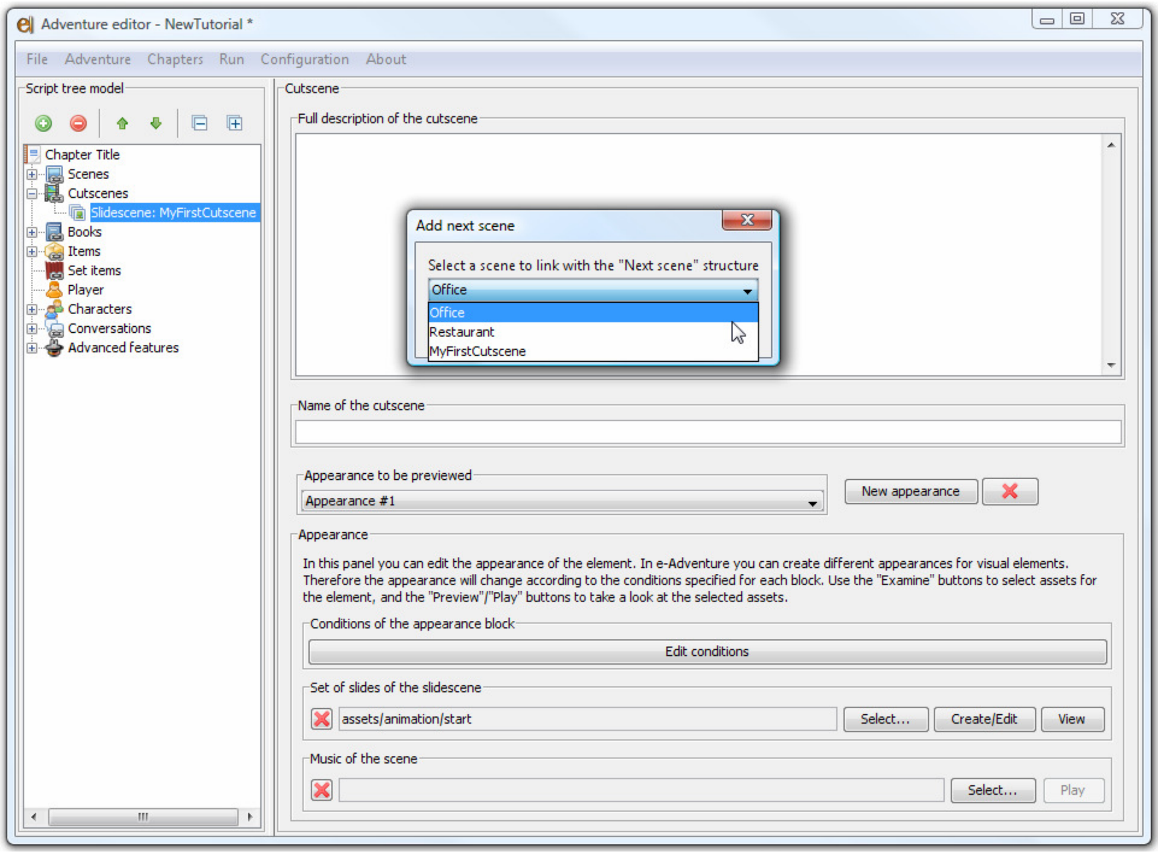
A priori the supported formats for videoscenes are MOV, MPG and AVI. However, you must take into account that there are lots of different codecs for all these formats, and some video clips may not work properly in <e-Adventure> due to the use of an unsupported codec, and this is independent of the format of the clip.

FILE FORMAT	CODEC	CODEC FAMILY	SUPPORTED
.AVI	DIVX 4	MPEG-4V	YES
.AVI	XVID	MPEG-4V	YES
.AVI	FFDS (VIDEO FLASH)		NO
.AVI	AVC	AVC	NO
.AVI	AVC	H264	NO
.AVI	MJPEG	MJPEG	NO
.AVI	mpg1		NO
.AVI	mpg2		NO
.AVI	WMV1	Win Media Video 7	NO
.AVI	WMV2	Win Media Video 8	NO
.AVI	DIVX3low	MPEG-4	YES
.AVI	S-Mpeg 4 V2	Microsoft MPEG-4 V2	YES
.AVI	MS-mpeg4 v1	MPEG-4V	NO
.AVI	DIVX5	MPEG-4V	YES
.AVI	MPEG-4 visual	MPEG-4V (fmp4)	NO
.AVI	MPEG-4 video	MPEG-4V (mpv4)	NO
.MPG	MPEG1 video	MPEG-V	YES
.MPG	MPEG2 video	MPEG-V	NO
.MOV	MPEG 4 visual	MPEG 4V	NO
.MOV	H 263	H 263	NO
.MOV	AVC	AVC	NO

The table includes much more tests for AVI video clips than for any other format. However, all the conclusions about the codecs supported in AVI format can be generalized to the other formats.

2.7.3. Linking cut-scenes to other scenes

As with scenes, cut-scenes can be linked to other scenes or cut-scenes. This is an essential behavior as if a cut-scene with no links is reached the engine will not know where to go next, therefore blocking the game. This process is very similar to linking scenes except that now we cannot talk about exits (cut-scenes are not navigable as scenes). You just need to do right-click on the node of the cut-scene you want to connect and select the option **"Add next scene"** if you want the player will move to a scene or cut-scene when the cut-scene is done, or the option "Add end scene" if you just want the game will finish after the cut-scene. For example, we will link our start slide-scene to the first scene of the game, so after the title the game truly begins. We just select "Add next scene" and after that select the scene you want to move to (in our case "Office").



In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

You can add as many next scenes as you find necessary. The engine will decide where to go checking the conditions of all the next scenes added. The first next scene satisfying all its conditions will be used.

3. Perfecting my first <e-Adventure> game: Advanced Features

Now you have the knowledge required to start creating <e-Adventure> games. However, <e-Adventure> supports further features, enhancing the capabilities of the <e-Adventure> games. In this section we will learn these features.

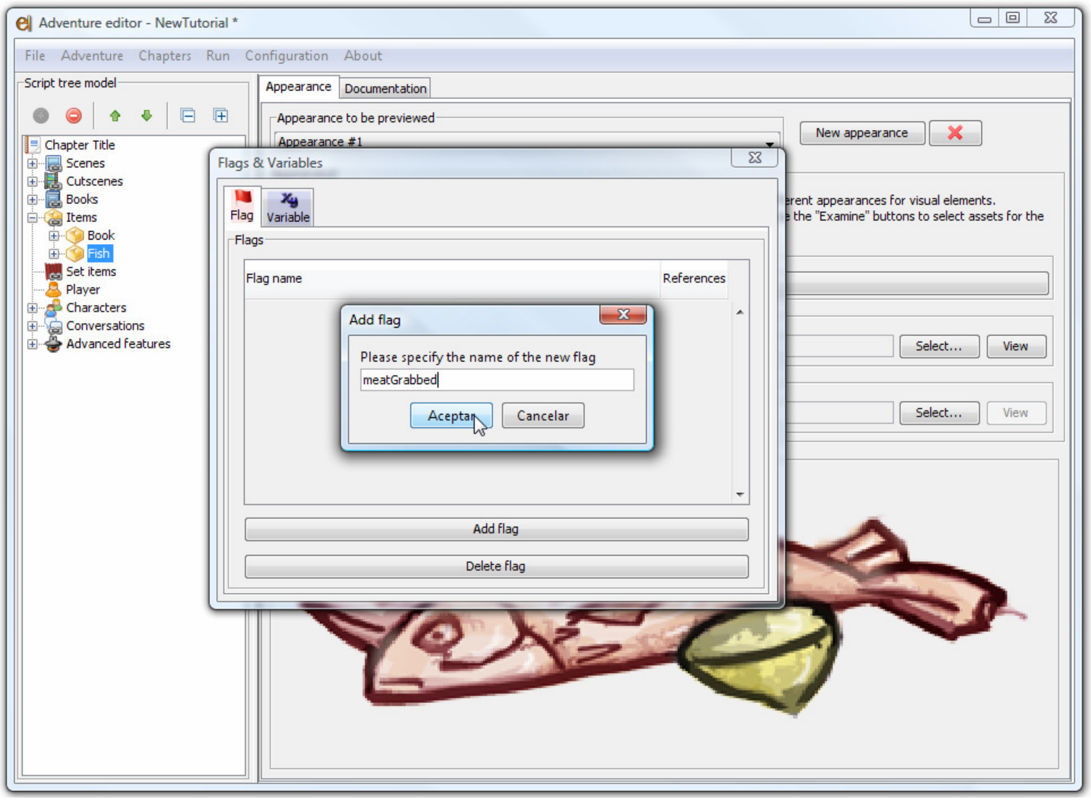
3.1. Conditions and effects

So far we have learned how to create and edit the elements of an <e-Adventure> game. However, a couple of items, characters and scenes do not make a game. To do so you need to drive the story of the game, achieving a narrative behavior. In <e-Adventure> the story is driven by ***flags*** and ***variables***. Flags are like "lockers" defined by the designer of the game which at any time can be evaluated as active or inactive (just as a locker can be locked or unlocked). Variables are entities that are assigned an integer value at any time. In this manner we can define flags and variables, and establish conditions on the value of these flags and variables. Let's go through this with an example.

3.1.1 A simple condition

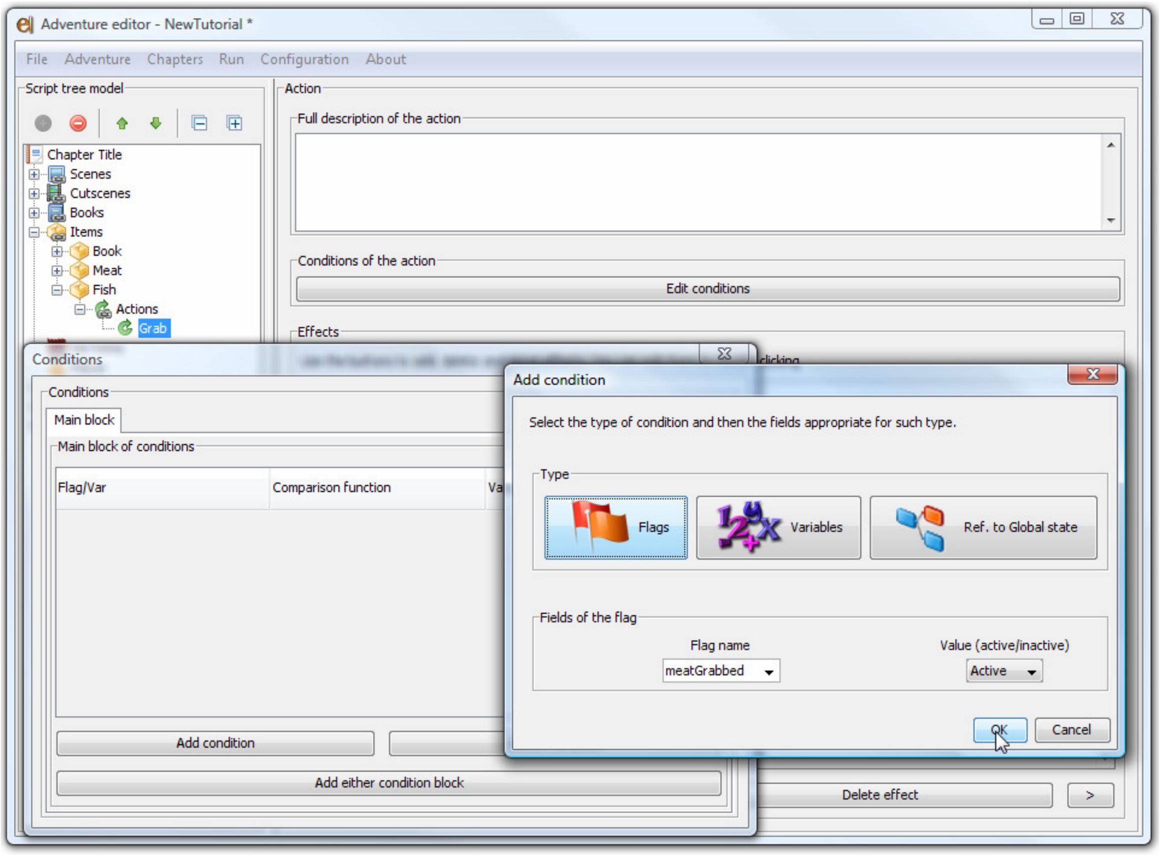
Guess we are defining a game in which we have two items: a piece of meat and a piece of fish. The point is we want that the player could grab the fish only after have grabbed the meat. In our adventure such condition is "translated" into the flags language: we must define a flag **"meatGrabbed"**, which is false all the time until the meat is grabbed. From that moment the flag will get its value set to true, and therefore the fish will be accessible to the player. Let's track the whole process from the beginning:

First, we need to add a new flag. To do that, go to the "Chapters" menu on the menu bar. There select the option "Edit chapter flags". You can perform this operation also pressing Ctrl+F. A dialog like the one below will be shown. Press on the button "Add flag" and type "meatGrabbed". Then press ok and a new flag with 0 references will be created.



You can use this dialog to add and delete flags (and variables). However, if you want to delete a flag first it must not have references at all. Besides, note that flags definition has some restrictions you must respect: no blank spaces are allowed and the first character must be a letter.

Once the flag has been created, you can make references to it. Then we can go to our fish item and add a new grab action. Remember: Expand the node of the item > Right click on "Actions" > Add "Grab" action. After that, press on "Edit conditions". You will get a dialog to edit the conditions. Press the button "Add condition" and you will be prompted to choose a flag and a state (active or inactive). If you select "active" the condition will be true when the flag is true. On the contrary, if you select "inactive" the condition will be true when the flag is false (by default all flags are inactive). In our case, we select "active" and the flag "meatGrabbed".



Note that the combo box containing the flags is editable. Therefore you can type the name of the flag. If this does not exist yet, it will be automatically created so you do not really need to create it previously.

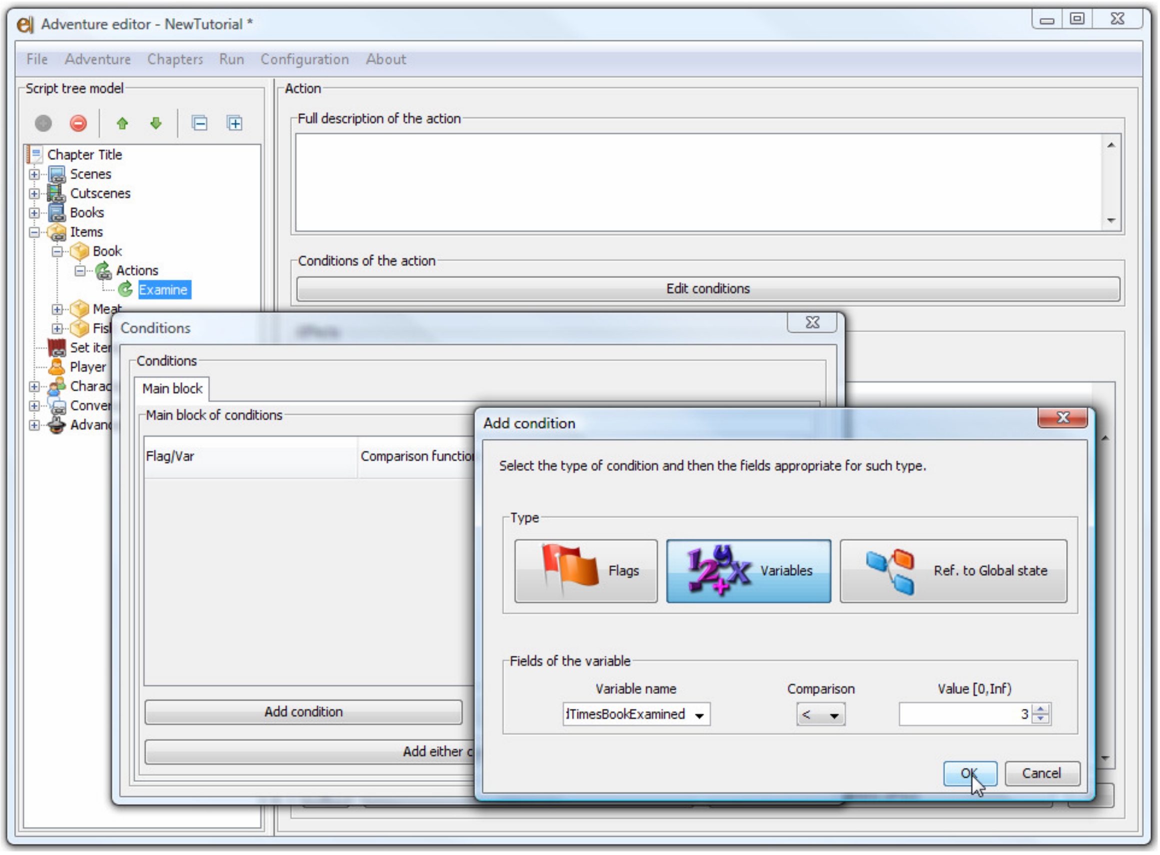
3.1.2. Variables

The flags system allows us to define the narrative flow in our videogames. However the use of flags becomes really arduous in several situations. For instance, guess we want to limit to three the number of times a player can carry out a specific action (be it to examine the item "Book"). To remember those numbers we should define a flag for each possible value (0, 1, 2 & 3 in this case).

For that reason <e-Adventure> also includes **variables**. The intuitive notion behind variables is that they are elements with a numeric value (0 by default). This value can be modified during the course of the games. To add a condition on the value of a variable we just need to perform a similar process to the definition of conditions on flags. Let's see it through the previous example.

To limit the number of times the item "Book" can be examined we create a variable, "TimesBookExamined", in order to keep the count of the times the book was examined. After that we add a condition to the "Examine" action of the book. The process is as follows:

First we create the new action "Examine" by right-clicking on the appropriate node of the tree. Then, on the right panel we get, just press the "Edit Conditions" button > "Add condition". The difference is that now we press the toggle button situated in the middle (**"Variables"**). The last step is to fill the fields that are displayed below with the next data: "TimesBookExamined" for the name of the variable, "<" as the evaluation function and 3 as value.



Add condition

Select the type of condition and then the fields appropriate for such type.

Fields of the variable

Variable name	Comparison	Value [0,Inf)
TimesBookExamined	<	3

3.1.3. Global states

Finally <e-Adventure> supports defining conditions over "global states of the game". Those "Global states" are just groups of conditions that are defined on a separate place to be reused more easily, and they can be referenced from different points of the games. They are especially useful when in a videogame there is a relevant state, depending on various flags and variables, which determines the flow of the narration in several points.

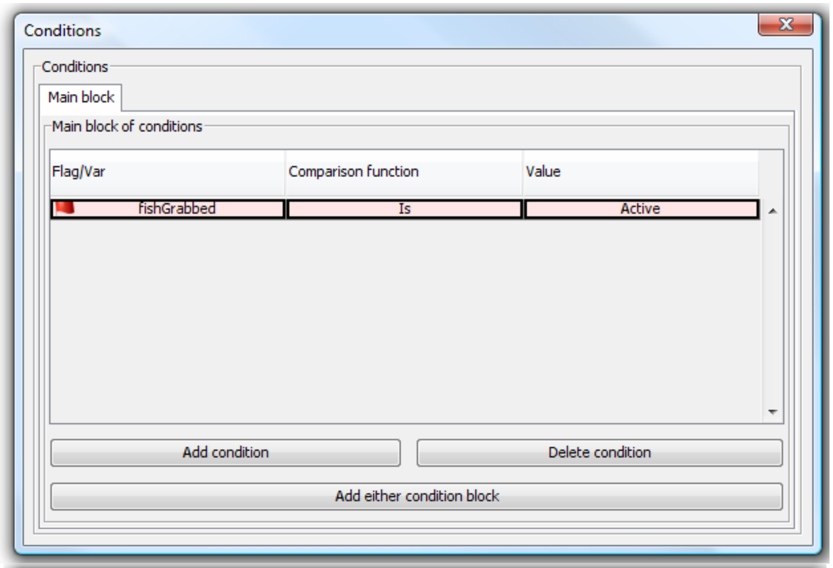
To create a new global state we must expand the node "Advanced features" of the left tree > Right click on the "Global states" node > Add global state. The panel shown on the right for the edition of the new global state is very similar to a conditions panel. The only extra restriction here is that in a Global state you cannot define a condition which refers to the Global state itself.

Once the global state has been created it is ready to be referenced in any conditions block of the game. It is a straightforward task: when a new condition is created, just press the third toggle button of the Condition panel ("Ref to a Global State") and select the id of the global state desired. This condition will be evaluated as true when the global state is true (that is, when its block of conditions is evaluated as true).

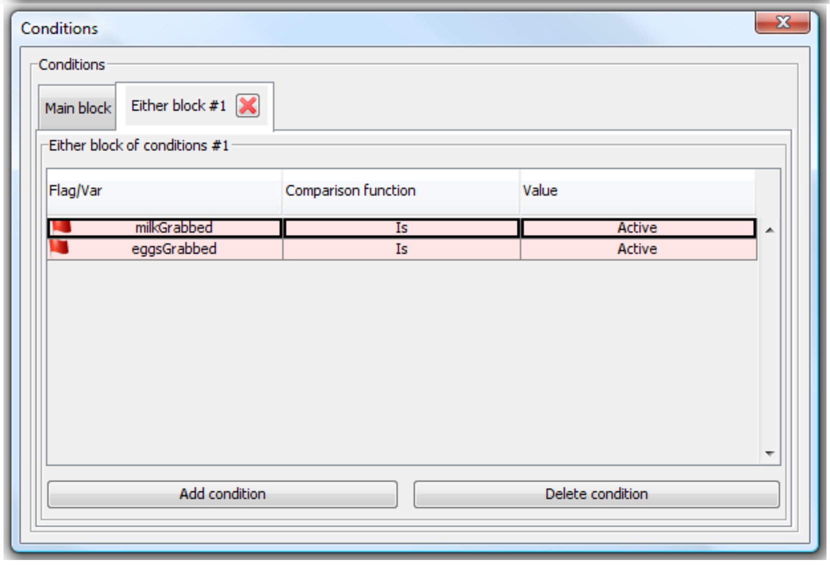
3.1.4. Either conditions blocks.

As you would have noticed, in the conditions dialog there are three buttons. Two of them are very obvious, as they allow you to add and delete conditions. The third one, labelled as "Add either condition block" will add a new tab next to the current one ("Main block"). Each tab is a block of conditions that must be true to consider the block is true. The condition will be true when at least one of the blocks is true (blocks are evaluated using the "or" logic operand). That is useful to create conditions depending on several flags. If some of the flags must be active or inactive at the same time, they will be located on the same block ("and" conditions). If only one of them needs to be active or inactive, you will add them in different blocks ("or" conditions).

Let's give an example. Guess we have an item *"Meat"* which *can only be grabbed if the item "Fish" has already been grabbed and the item "Milk" or the item "Eggs" has been grabbed (or both).* If when these four ítems are grabbed the flags "meatGrabbed", "fishGrabbed", "milkGrabbed" and "eggsGrabbed" are activated respectively, the condition which models this behaviour will have two either blocks and will look something like this:



Flag/Var	Comparison function	Value
fishGrabbed	Is	Active



Flag/Var	Comparison function	Value
milkGrabbed	Is	Active
eggsGrabbed	Is	Active

3.1.5. What you can do with conditions

So we know how to define conditions. You can use them in almost all the contexts, changing the behavior of the game

- *For actions (items and active areas)*

- *For conversations (characters)*

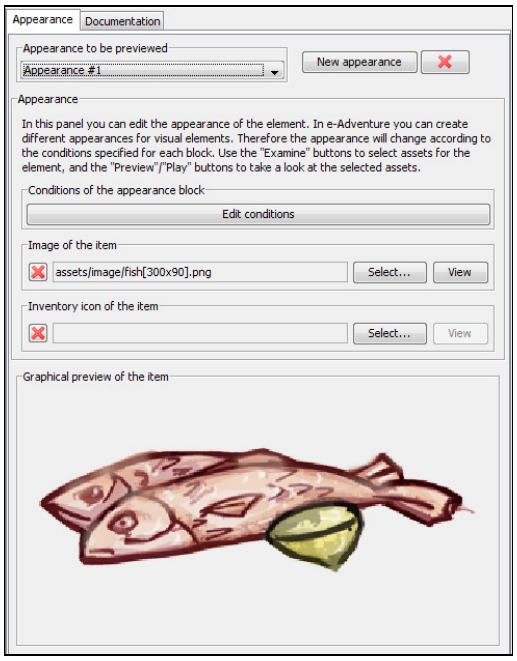
- *For scenes*:

- *For barriers* (to force a barrier to block or unblock the player go through it depending on a group of conditions).

- *To change where you will go when pressing on an exit.* (You can define several next scenes for an exit, and the first whose conditions are satisfied will be picked)

- *For item, set item and character references.* Hence the elements present in a scene will be allowed to change during the game.

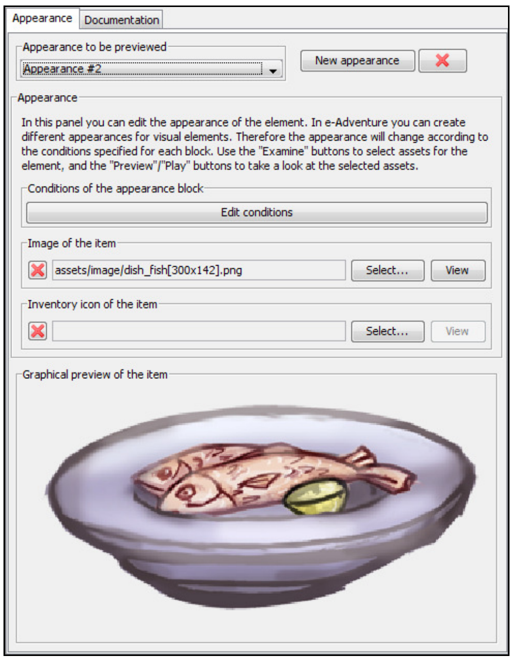
- *To adapt the appearance of an element.* All the elements in <e-Adventure> allow you to create different "Appearance Blocks". The resources used for each appearance block will be different. Then when an element is displayed the appearance used will be the first which conditions are satisfied. For instance, we can do so to put our fish on a plate in a certain moment of the game by using two different images:


Appearance
Documentation
Appearance to be previewed
New appearance
Appearance
Conditions of the appearance block
Edit conditions
Image of the item
Select...
View
Inventory icon of the item
Select...
View
Graphical preview of the item

Appearance #1

In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

assets/image/fish[300x90].png


Appearance
Documentation
Appearance to be previewed
New appearance
Appearance
Conditions of the appearance block
Edit conditions
Image of the item
Select...
View
Inventory icon of the item
Select...
View
Graphical preview of the item

Appearance #2

In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

assets/image/dish_fish[300x142].png

3.1.6. Activating and deactivating flags: Effects

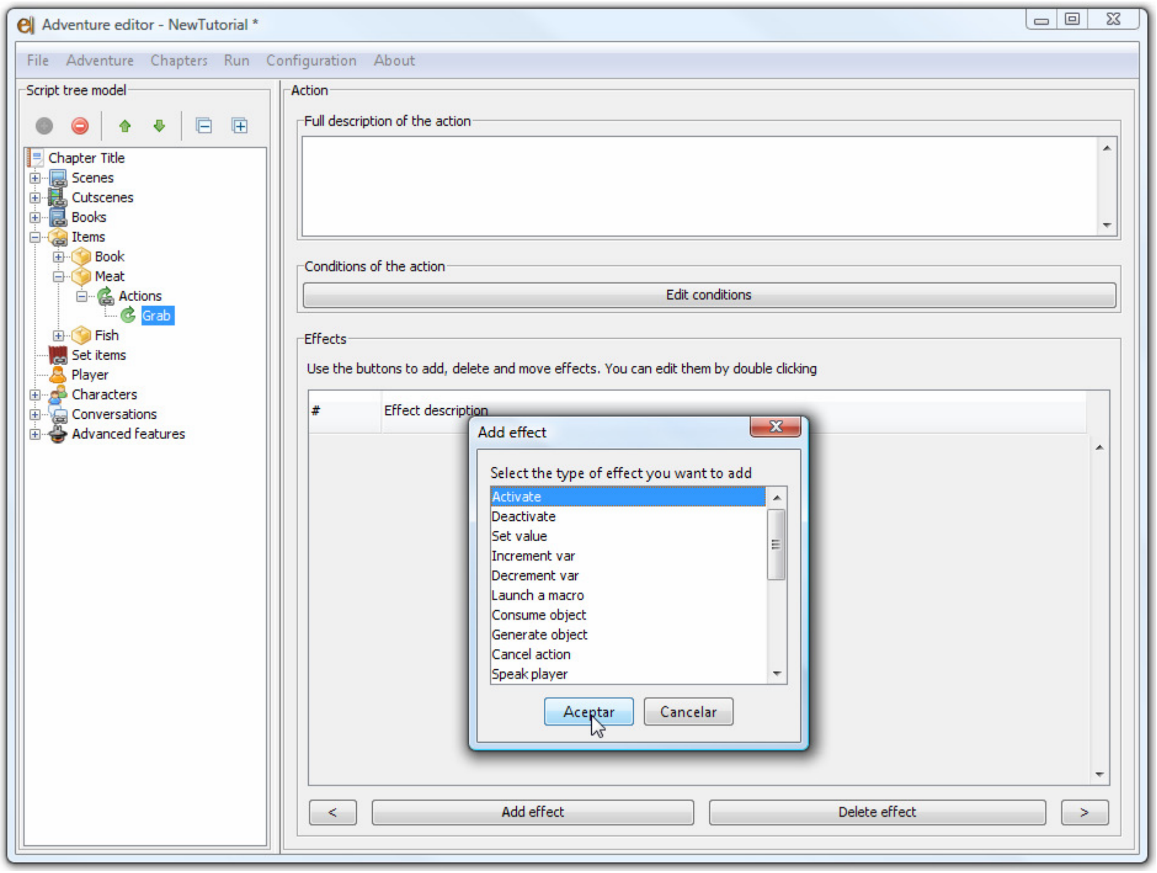
As you can see, defining conditions in the games is useful tool. But, how do we make the flags change? Establishing conditions on their own will not produce any effect as they depend on flags which will never change. In <e-Adventure> you can activate or deactivate flags by using effects. You can define those effects in the next situations:

- *For actions (items and active areas)*. When the action is carried out its effects will be executed.

- *For conversations (characters)*. You can define effects in any node of a conversation. Thus when the node is reached those effects will be executed.

- *For next scenes*. You can produce effects when the current scene changes. In that case you can define effects, which will be executed before the scene changes, and post-effects, which will be executed once the scene has changed.

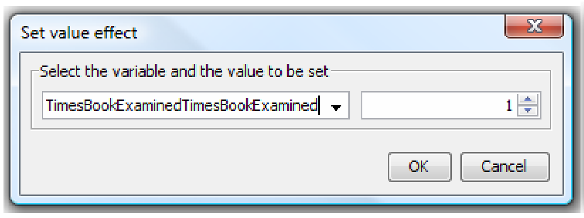
In our case, if we want to get the flag "meatGrabbed" activated after the meat has been grabbed we must add a grab action for the item and edit its effects:

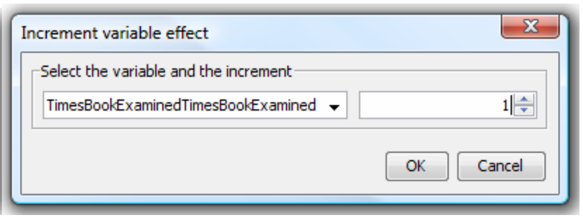


To do that, select the node of the action on the tree. Then press on "Add effect" and select "Activate". Then you will be prompted to choose which flag you want to activate. There select "meatGrabbed".

3.1.7. Giving value to variables.

As flags can be activated or deactivated, variables can be assigned a value as well. This can be done using the effects **"Set value"**, **"Increment var"** and **"Decrement var"**. The first one will allow us to set the exact value (0 to infinity) that the variable will take. The next two effects add or subtract the given positive value to the variable.


Set value effect
Select the variable and the value to be set
TimesBookExaminedTimesBookExamined
1
OK
Cancel


Increment variable effect
Select the variable and the increment
TimesBookExaminedTimesBookExamined
1
OK
Cancel

3.1.8. Other effects.

Effects are used when you want to activate or deactivate flags. However, <e-Adventure> supports much more kinds of effects. Here you can find all the effects supported along with a description for each one and suggestions about how to use them

Effect	Description	Uses
Consume object	Removes an item from the scene.	A typical use for this effect is to make objects disappear when they are grabbed.
Generate object	Puts an item in the inventory	Typically used when objects are grabbed
Cancel action	Cancels the default action	Use it when you do not want an item to be grabbed for grab actions, or to avoid an item to be consumed when it is given to a character.
Speak player	Makes the player to say a single line	You can use this kind of effects when an action is forbidden
Speak character	Makes a character to say a single line	You can use this kind of effects when an action is forbidden
Trigger book	Displays one of the books defined in the game	This effect can be used to provide some information for a limited period of time
Play sound	Plays the selected sound	Can be used to produce sound effects for actions as those sounds are played once
Play animation	Plays a new animation	Can be used to trigger sets of slides.
Move player (Third person mode only)	Makes the player to go to the selected position (x, y)	A typical behavior in adventure games: characters and players move while speaking
Move character	Makes the selected character to go to the selected position (x, y)	
Trigger conversation	Triggers the selected conversation	You can use this effect to provide interactive guidance at some points in the game
Trigger cutscene	Triggers the selected conversation	A video or set of slides can be played to provide information or explanation at some points in the game
Trigger scene	Changes the current scene	Useful since it allows to change the scene without using an exit
Effect with probability	Depending on a given probability the game engine decides the effect to launch from a list of two effects	Allows pseudo-random behaviours in the adventures.
Trigger last scene	Changes the current scene,	Useful to implement

	coming back to the last visited scene.	navigational environments when you cannot know which is the next scene to visit a priori
Launch macro	Launches a macro (set of effects)	Facilitates the reutilisation of the effect blocks.

3.1.9. Macros.

Finally <e-Adventure> supports the creation of groups of effects (macros). These groups of effects are given a unique ID so you can make references to them at any point, which eases the maintenance of the games and the reuse of the effects. To create a macro go to the Advanced Features node > Macros > Right click > Add macro. Then type the ID and the macro will be created. There you will be able to add and remove effects as for any other effects block of the game. You have only one restriction: you will not be able to add a Launch Macro effect to trigger the same Macro you are editing!

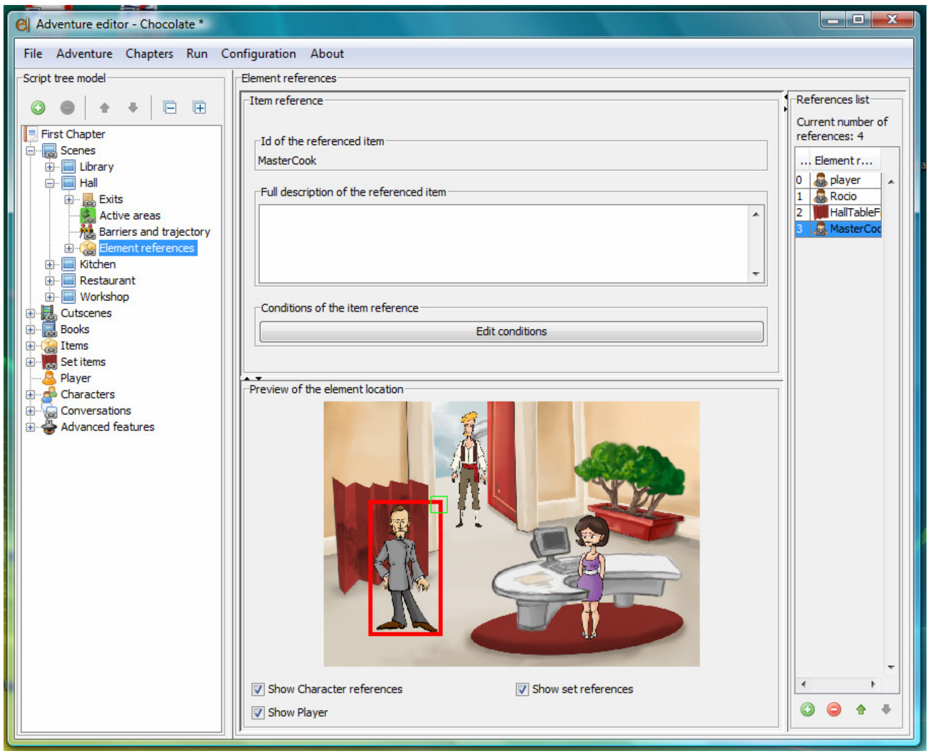
To launch the macro, just use the "Launch macro" effect from any other effects block in the game.

3.2. Organizing the elements in the scene: Layers.

All along this document you have learnt to instantiate items, set items and characters in the scenes. Usually this process consists mainly of setting the place where you want to put the element on the scene (that is the x, y coordinates) and the size of the element. However, in <e-Adventure> you can also edit the "depth" of the element.

For example, let's suppose that we add a table in the scene (as set item), and we want to see the player behind. It means, when the player collides with the table, it must be painted above the player. You can choose which elements will be in front of and behind others, providing the depth of the element in the scene ("z" coordinate). From now on we will name this the **"layer"**, defined as the relative position of the element in scenes in regard to other elements.

To select the element layer, you have to go to scene node in the left tree, expand it, and select "***Element references***". When you do it, the next panel will appear:



In that panel you will find tree sub-panels. You can adjust the portion of screen each panel occupies by dragging and dropping the split bars. Besides, you can hide and show them completely by double clicking on them.

Information and conditions

When you select a reference, in the preview panel (below) or the references list (right), this field appears on the panel on top, filled in with the information of the reference. The first field is the name, and it is not editable. To change it, you must go to the element edition panel. Next you will find the reference documentation. You can fill it in freely (please note the doc here is not the doc of the element but of the concrete reference). The last one is the "***Edit conditions***" button, which allows you to add restrictions to the reference by flags, variables or game states (for more details see section **3.1**). With conditions you can hide and show an element only in some desired situations.

Elements preview

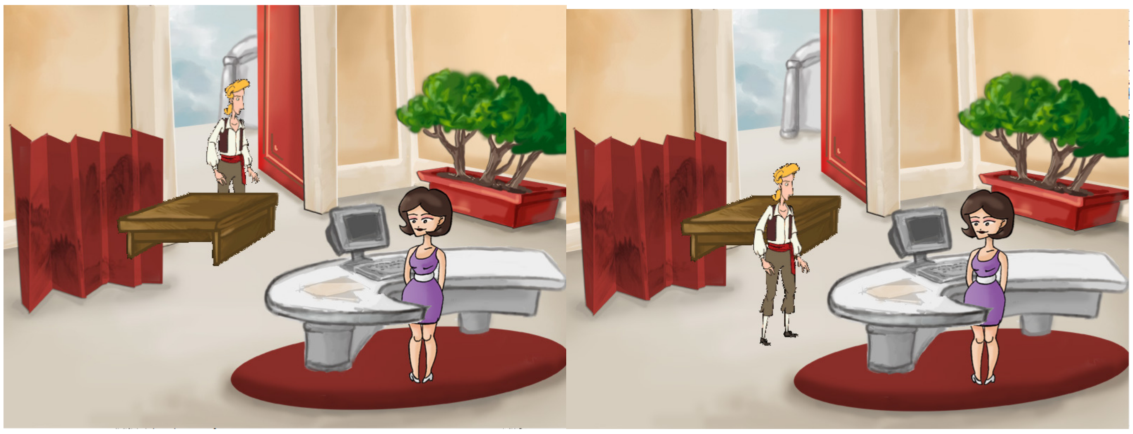
This sub-panel shows how the elements are placed in the scene, previewing the elements with the position, size and depth they will have in the game. To edit how the elements are organized, you can modify the position by dragging and dropping along the scene. Also you can modify the element's size by dragging and dropping the green square you will find on one of the corners of the element.

In addition, at the bottom of that panel you will find several check boxes (**"*Show set items references*"**, **"*Show character references*"**, etc.) to select what kind of elements must be included in the preview (items, set items, characters and player). If some elements disturb your vision when you are editing the scene, you can remove it temporally. It is important to emphasize that this changes only affects at this panel, all added references will be shown in the game.

References list

The references list you find on the right of the panel shows all the references and the player of the scene ordered according to their layer. The idea is that the lower an element is in the table, the more they are at the front. That is, elements at the bottom of the list (with lower depth) are painted before the other elements. In short, if you want an element A to be in front of an element B make sure that element A has a greater layer (for instance 5) than the element B (for instance 2).

In this panel you can add or remove references using the (+) and (-) buttons. To change the layer of the elements (i.e. the order in which they must be painted) use the (↑) and (↓) buttons.



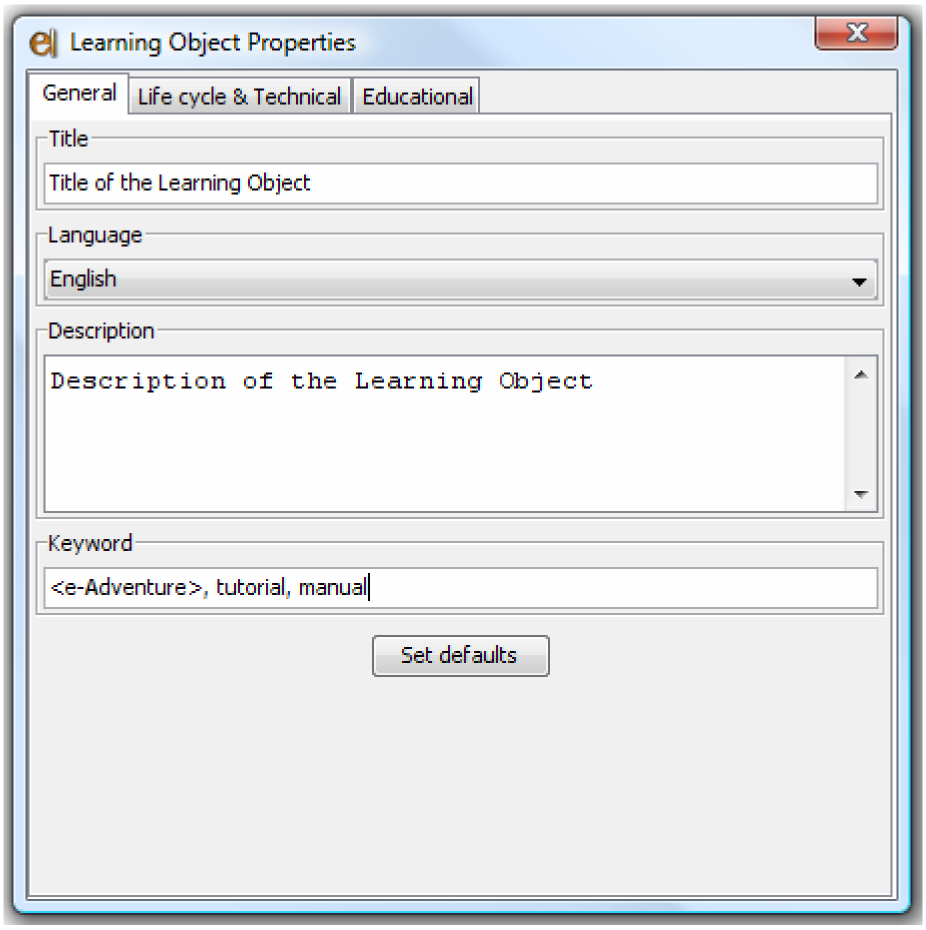
Another interesting feature here is that the z coordinate of the player can be adjusted automatically, without taking into account its layer. This is interesting in situations where the player must be painted behind an element sometimes and in front of it other times. The typical example would be something like the image above (when the player moves forward must be painted in front of the table, otherwise must be painted behind).

To get this effect you should deactivate the player's layer. Select the scene node and in the main panel, select the information tab. Deactivate the option "*Allow to choose player's layer*" that it is placed at the bottom and the layer of the player will not be editable any more. Then the game engine decides the z position where to paint the player comparing the coordinate y of the player and the other elements (the greater y is painted on top).

3.3. Exporting the games as Learning Objects.

An interesting feature of the <e-Adventure> platform is that they allow game authors to package their adventure games as Learning Objects (LO). A LO is a self-contained package which can be deployed in a standard-compliant Learning Management System (LMS), such as Moodle, Sakai, WebCT-Blackboard, etc. In this manner <e-Adventure> videogames can be embedded in online courses just as a simple HTML document, which eases the delivery of the games to the students.

Before you export an adventure is as LO you can annotate some meta-data describing the LO. To access and edit these data go to the File Menu > Learning Object properties. Then a dialog like the next one will be displayed:



Title
Title of the Learning Object

Language
English

Description
Description of the Learning Object

Keyword
<e-Adventure>, tutorial, manual

There are several fields you can edit. Those are:

“General” data Tab

- *Title*: Title of the adventure.
- *Language*: Language used to write the game (English or Spanish only).
- *Description*: General description of the game and the plot of the game.

NOTE: Both the tile and the description can be filled automatically using the general data of the adventure. To do this, just press on the button **“Default Values”.**

- *Keyword*: These keywords can be used to index the LO in a LO repository.

“Life Cycle & Technical” data Tab

- *Version*: Game version (do not mistake with the <e-Adventure> version.
- *Min version:* This value is not editable, and indicates the minimum version of the game engine required to launch the game.
- *Max version:* This value is not editable, and indicates the maximum version of the game engine which can run the game.

“Educational” data Tab.

- *Intended end user role*: Role of the person to which the game is oriented. Supported values: Teacher, Author, Learner or Manager.
- *Semantic density:* Please, note that this field is a bit tricky, as its meaning does not match to the intuitive notion of "Semantic density". The real purpose of this field is to indicate how much the game covers the domain knowledge. The accepted values are: Very low, low, medium, high and very high.
- *Learning resource type:* It can take one of the next values: exercise, simulation, exam, self assessment or lecture.
- *Context:* Area of the educational system to which the game is oriented. Accepted values are: School, Higher education, training or other.
- *Difficulty*: The difficulty of the learning resource, from very easy to very difficult.
- *Interactivity level*: Describes how much interactive this learning resource is. It can take values from very low to very high, but we recommend setting this value to high or very high.
- *Interactivity type*: Accepted values are: active, explosive or mixed.
- *Description*: This field has nothing to do with the general description. It must describe the learning goals and the educational application context of the videogame.
- *Typical age range:* Typical age of the students to which the game is oriented, for instance 16-25.
- *Typical learning time:* Estimated time in hours and minutes to complete the whole game.

All these fields are defined following the IEEE LOM (Learning Object Metadata) standard. If you have any doubt, you can access it following this link:

http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf.

You do not need to fill these fields if you don't want. The game will be exportable anyway. The results of the exportation are a ZIP file which will be saved wherever we decide in our file system. However, by default the LO exported will be stored in the *Exports* folder of the EAD_HOME.

The process to export the game as a LO is very simple. Go to the File Menu > Export > As Learning Object. Then you will be prompted to type the name of the learning object, your name and the name of your organization and a last parameter which indicates how to execute the game on the student's browser.

The first field is not especially relevant, you can leave the default LO name. It is just used by the LMS to distinguish all the learning objects it contains.

On the other hand the author and organization names are compulsory due to security reasons. Those parameters must be at least 6 characters long and must not contain blanks. With both parameters the editor will generate a certificate to sign digitally the game so an Internet browser can execute it.

Finally, the check box labelled as "The applet runs inside the browser" indicates if the game must be embedded into the Web browser (if activated) or if the game must run on a separate window on top (if the checkbox is deactivated).

When these fields are filled press OK and then select the folder where to export the game and the file name.

3.4. Exporting games as self-contained, Java ARchive files (JAR).

Another interesting feature of <e-Adventure> is that it supports the exportation of the games as Java ARchive file. In this manner both the <e-Adventure> engine and the game as packaged as a single .JAR file which can be executed on your system by doing a double click. The only requisite is to have installed the Java Runtime Environment (at least version 1.6.0_02) on your system. This eases the distribution of the videogames when there is not a LMS to centralize the process.

The process to export the game as a JAR file is very similar to the one described in the previous section. Go to the *File Menu > Export project > Export as standalone jar-based game*. Select the target folder and the file name, and after a while the jar file will be created.

3.5. Adaptation and assessment profiles

Two of the most popular trends in online learning nowadays are how to adapt the learning experience to the needs of each student (known as adaptation) and how to track and assess that learning experience. Thus <e-Adventure> supports both features.

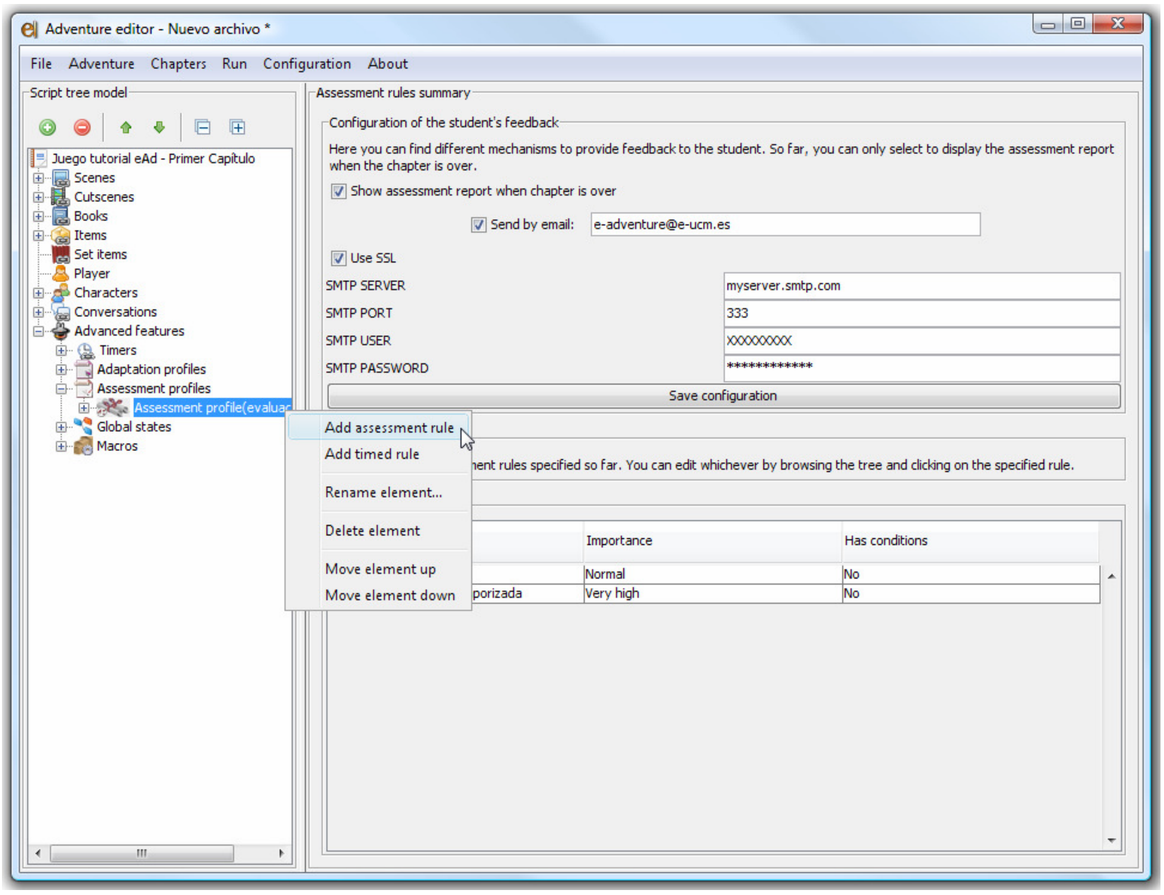
Those characteristics are defined through *Adaptation and assessment profiles*. Each profile is a set of rules defined as conditions over the set of flags and variables of the game. When those conditions are satisfied the rule triggers some effects, oriented to adapt the learning experience in the first case and to produce and assessment report in the second case.

The key behind the edition of the profiles is that you can have several assessment and adaptation profiles for each game for diverse audiences and contexts. Besides each chapter can have its own assessment and/or adaptation profile. To edit all these features you must go to the "Advanced features" node on the left tree.

3.5.1. Assessment profiles.

Let's start by creating a new assessment profile. Expand the "Advanced features" node and then go to the "Assessment profiles" node > Right click > Add new assessment profile. The editor will ask the name of the xml file which will store the profile (profiles are stored separately to the definition of the chapters). Then we are ready to edit the profile.

As the next figure depicts, the main panel of the profile has a section on the top where we can tell the game engine to show the assessment report to the student as feedback when the game is completed. In addition, when this option is selected we can activate the next option, which will tell the engine to send the report via e-mail (usually to the teacher). Due to privacy reasons, the assessment report cannot be sent via e-mail if the student does not agree with it.



Assessment rules summary

Configuration of the student's feedback

Here you can find different mechanisms to provide feedback to the student. So far, you can only select to display the assessment report when the chapter is over.

☑ Show assessment report when chapter is over

☑ Send by email: e-adventure@e-ucm.es

☑ Use SSL

SMTP SERVER	myserver.smtp.com
SMTP PORT	333
SMTP USER	XXXXXXXXX
SMTP PASSWORD	************

Save configuration

…ent rules specified so far. You can edit whichever by browsing the tree and clicking on the specified rule.

	Importance	Has conditions
	Normal	No
porizada	Very high	No

To send e-mails the game engine will need the settings of your SMTP server. The two first fields, *SMTP Server* & *SMTP Port* must be provided by our e-mail provider (e.g. Gmail, Hotmail, Yahoo, a corporate e-mail server, etc.). The next two fields usually match to the username and password we use to access our e-mail mail box.

To add new rules we only have to right-click the node "Assessment report". There are two kinds of assessment rules: Assessment rules and timed rules.

Assessment rules

An assessment rule has the next fields:

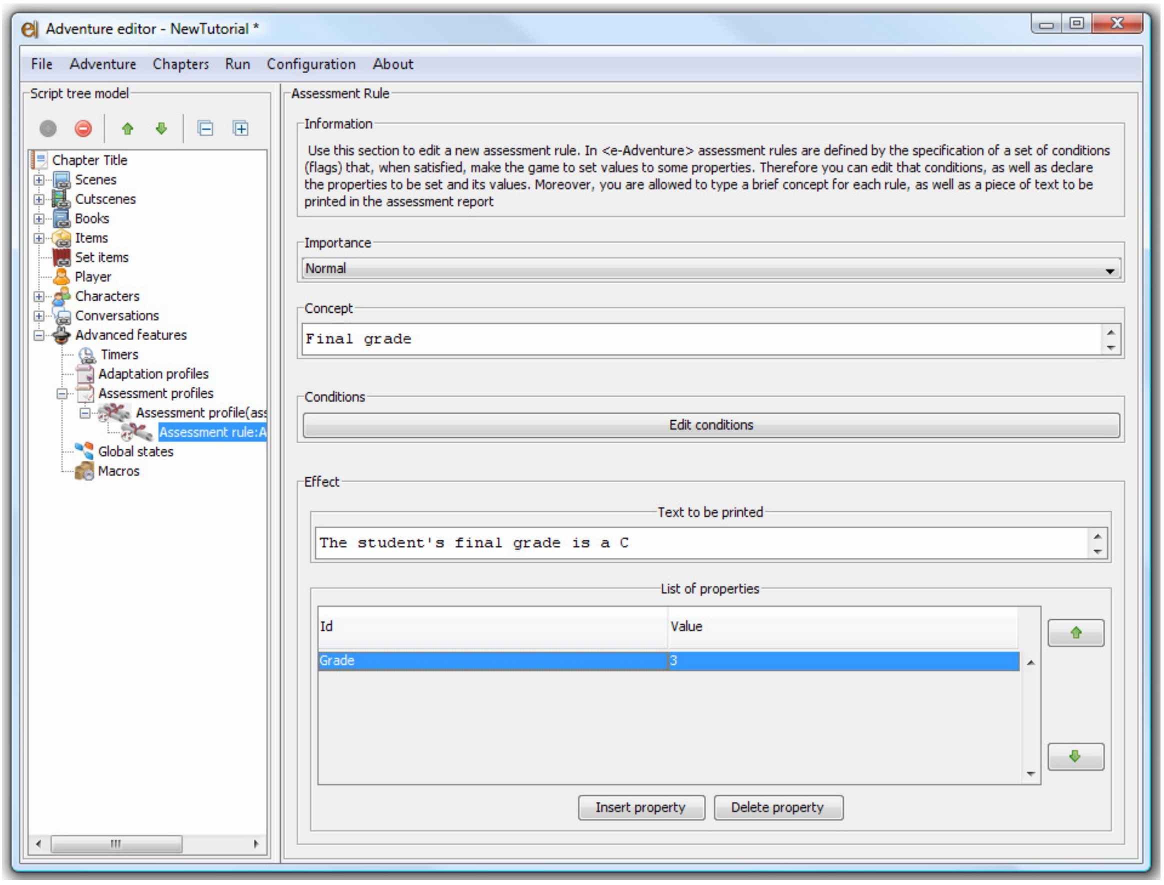
Importance: By default it will take the value "Normal". Other accepted values are: *Very low, low, normal, high* and very high.

Concept: Short description of the concept which tries to evaluate the rule. For instance, "Basic cooking skills".

Conditions: The set of conditions that must be satisfied in order to trigger the effects of the rule. The typical use is to define the conditions of the rule depending on a flag which is activated when the student performs an action.

Text: This text will be written directly on the assessment report, just below the Concept (which is written in bold format). For instance, here we could type "The student does not marriages properly the meat and the sauce".

Properties: Each property is a pair (attribute, value). When the conditions are satisfied the attributes are given such values. The typical use is to compute the final grade of the student. These properties will not be written onto the report, but sent to the LMS if possible so the server can process and attach the results to the profile of the student.



Use this section to edit a new assessment rule. In <e-Adventure> assessment rules are defined by the specification of a set of conditions (flags) that, when satisfied, make the game to set values to some properties. Therefore you can edit that conditions, as well as declare the properties to be set and its values. Moreover, you are allowed to type a brief concept for each rule, as well as a piece of text to be printed in the assessment report

Importance: Normal

Concept: Final grade

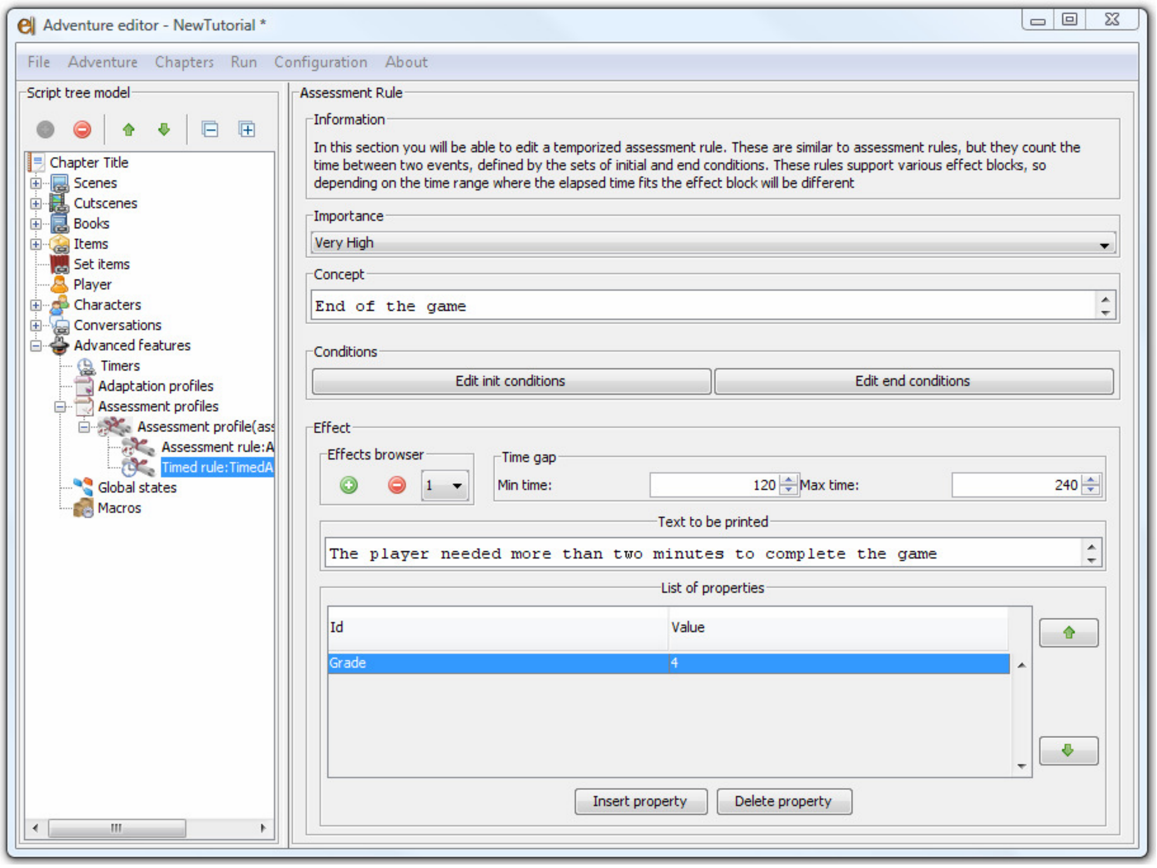
Text to be printed: The student's final grade is a C

Id	Value
Grade	3

Timed rules

Timed rules allow the evaluation of the students taking into account the time elapsed. When the **initial conditions** are satisfied a timer is launched (starts counting), and it is halted when the **end conditions** are satisfied. Then the elapsed time can be computed and launch a set of assessment effects according to some time ranges.

Therefore a timed assessment rule can have many effect blocks. Each block has a time range which determines if the effects of that block must be triggered. That is, if the elapsed time is within the range, the effects are triggered. Otherwise the next effect block (if any) is checked. If the elapsed time is not within any of the ranges, the rule has no effect on the evaluation of the student.



Assessment Rule

Information: In this section you will be able to edit a temporized assessment rule. These are similar to assessment rules, but they count the time between two events, defined by the sets of initial and end conditions. These rules support various effect blocks, so depending on the time range where the elapsed time fits the effect block will be different

Importance: Very High

Concept: End of the game

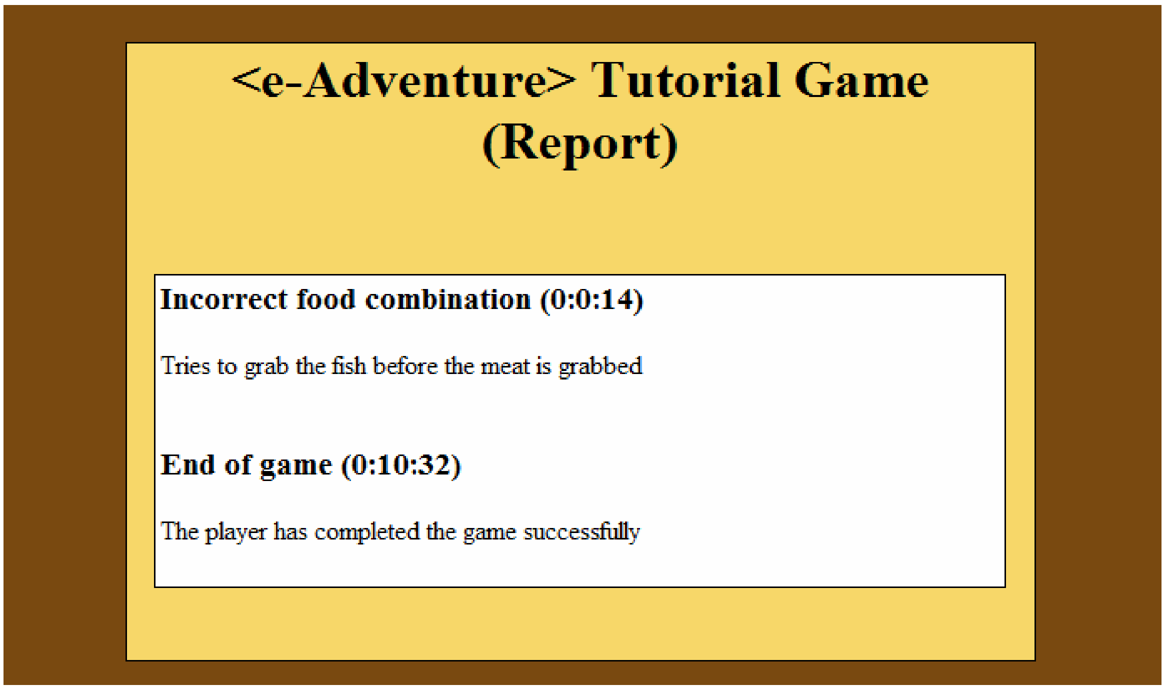
Effect — Time gap: Min time: 120 Max time: 240

Text to be printed: The player needed more than two minutes to complete the game

List of properties

Id	Value
Grade	4

All the effects of the different assessment rules are written onto two assessment reports. The first report, which is XML-based can be sent to a LMS while the second is HTML-based and can be used as a source of feedback for the student or as a report for the instructor (see figure below).



<e-Adventure> Tutorial Game (Report)

Incorrect food combination (0:0:14)

Tries to grab the fish before the meat is grabbed

End of game (0:10:32)

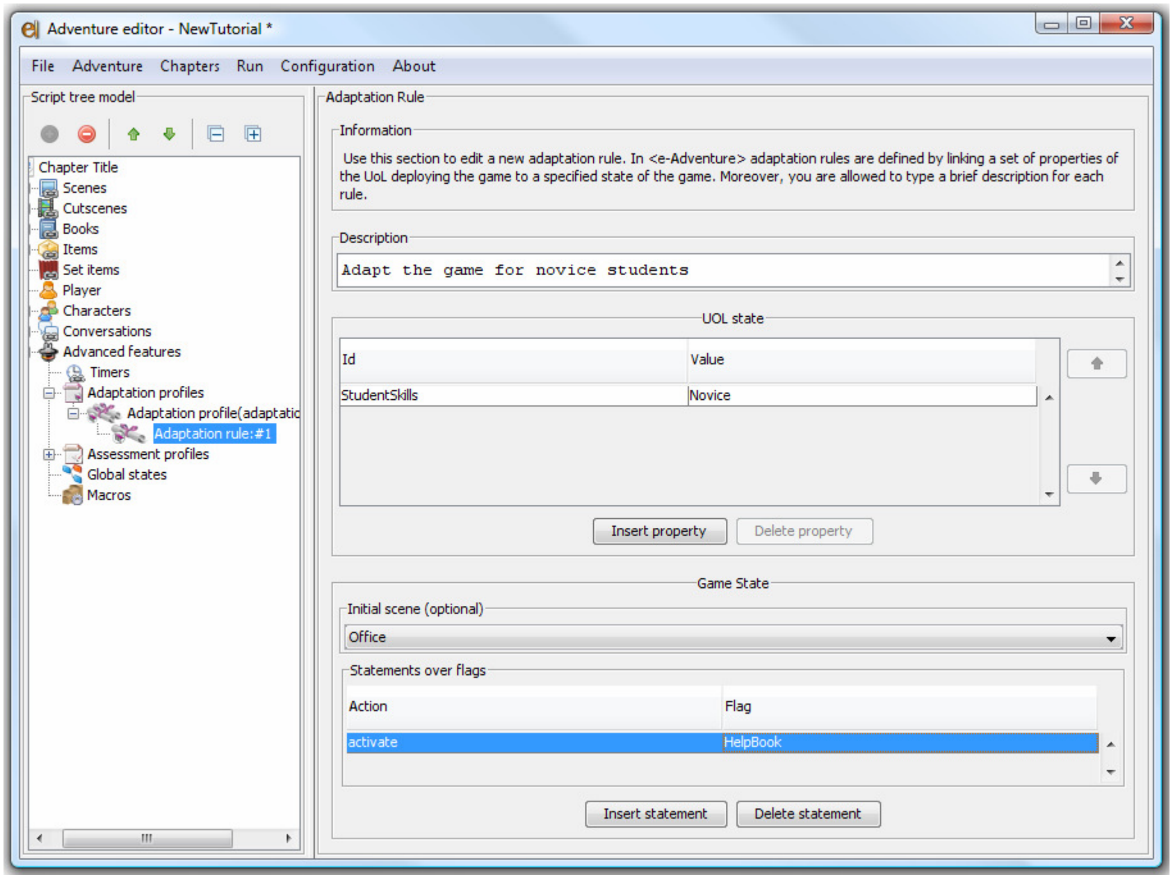
The player has completed the game successfully

3.5.2 Adaptation profiles

The adaptation profiles are generated analogously to the assessment profiles. This allows the automatic adaptation of the learning experience according to a set of "external properties", which are delivered from the LMS (usually when the game is launched). The game engine checks the status of these "external conditions" at the beginning of the game and when the values of the properties match the ones defined in the adaptation profile some flags are activated or deactivated to adapt the game accordingly.

A typical example of adaptation is the one in which the difficulty of the game is gauged according to an initial categorization of the skills of student (e.g. novice, intermediate, expert). Then we can associate with the game editor that if the property "KnowledgeLevel" of the LMS has the value "Novice" the flag "NeedsBackgroundReading" is activated and as a consequence a special book is unlocked and set into the inventory.

An adaptation rule looks like this:



Use this section to edit a new adaptation rule. In <e-Adventure> adaptation rules are defined by linking a set of properties of the UoL deploying the game to a specified state of the game. Moreover, you are allowed to type a brief description for each rule.

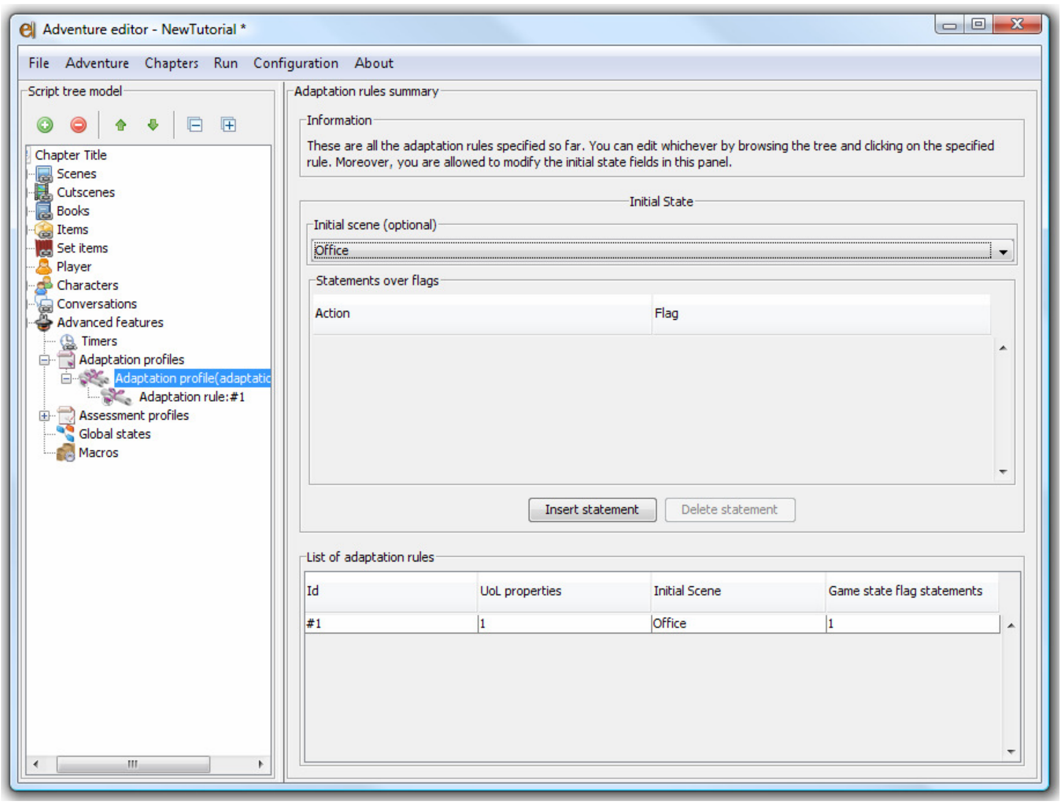
Description: Adapt the game for novice students

Id	Value
StudentSkills	Novice

Initial scene (optional): Office

Action	Flag
activate	HelpBook

Besides you can edit the initial state of the game using the profile:

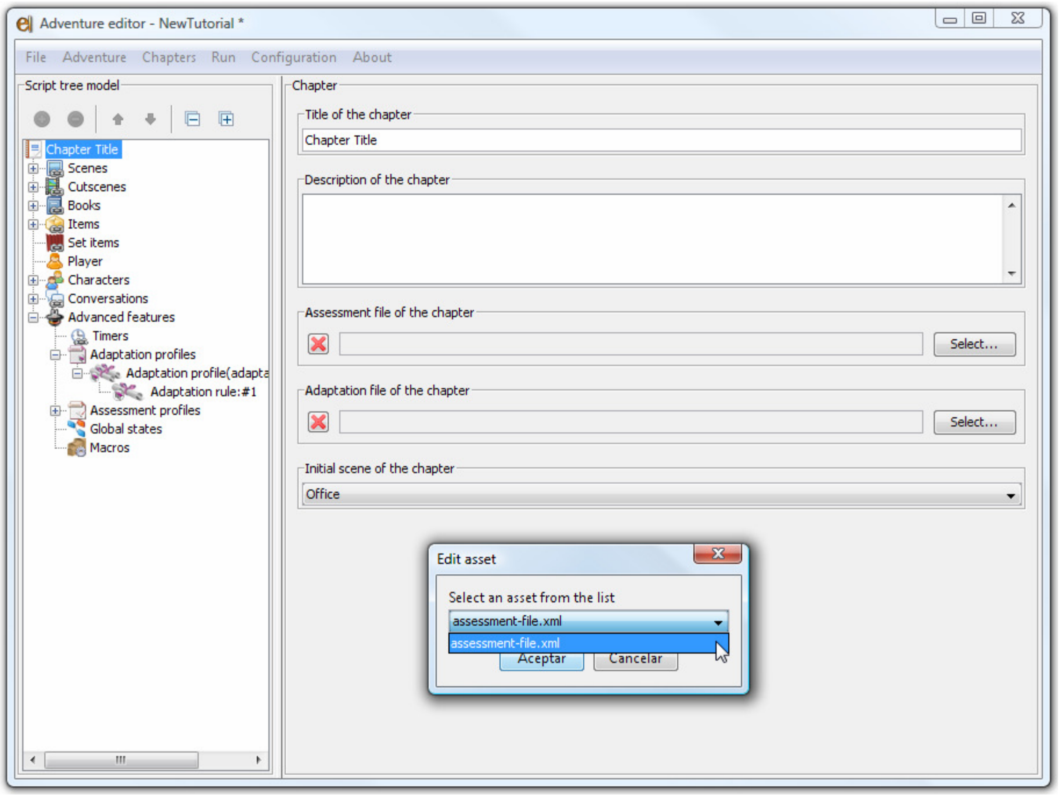


These are all the adaptation rules specified so far. You can edit whichever by browsing the tree and clicking on the specified rule. Moreover, you are allowed to modify the initial state fields in this panel.

Id	UoL properties	Initial Scene	Game state flag statements
#1	1	Office	1

3.5.3 Set the active profile for the chapter

The last thing you need to do when you are done editing an assessment or adaptation profile is to set the active profile on the chapter panel. To do this, select the root node of the left tree, and use the "Select…" buttons you will find below the chapter's title and description text areas.

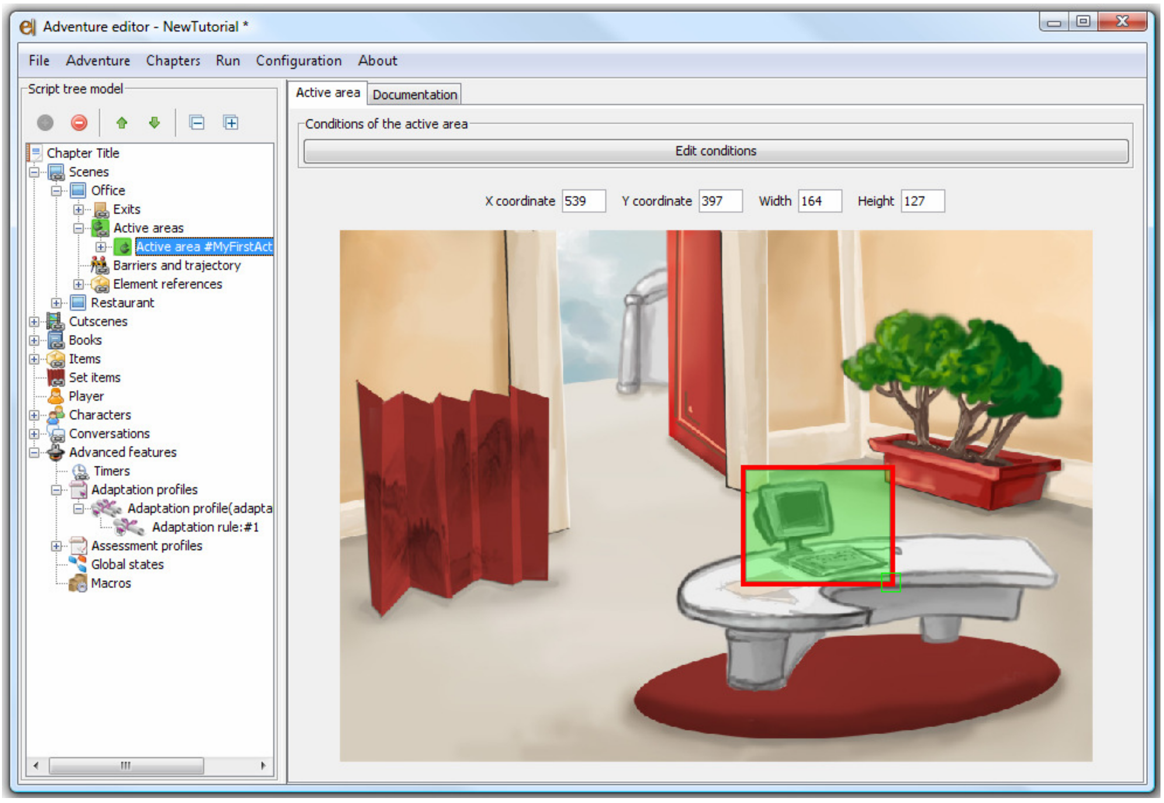


Title of the chapter: Chapter Title
Description of the chapter:
Assessment file of the chapter:
Adaptation file of the chapter:
Initial scene of the chapter: Office

3.6. Active areas

The active areas are rectangular portions of the background of the scene with which the player can interact. They are very useful if we have rich background images with several items embedded on it, or if the game is developed using photos of real scenarios.

Regarding the interaction with active areas, those are just as items which are embedded into a single scene. Their area can be delimited as for exits, and contains the same descriptions and actions that items.



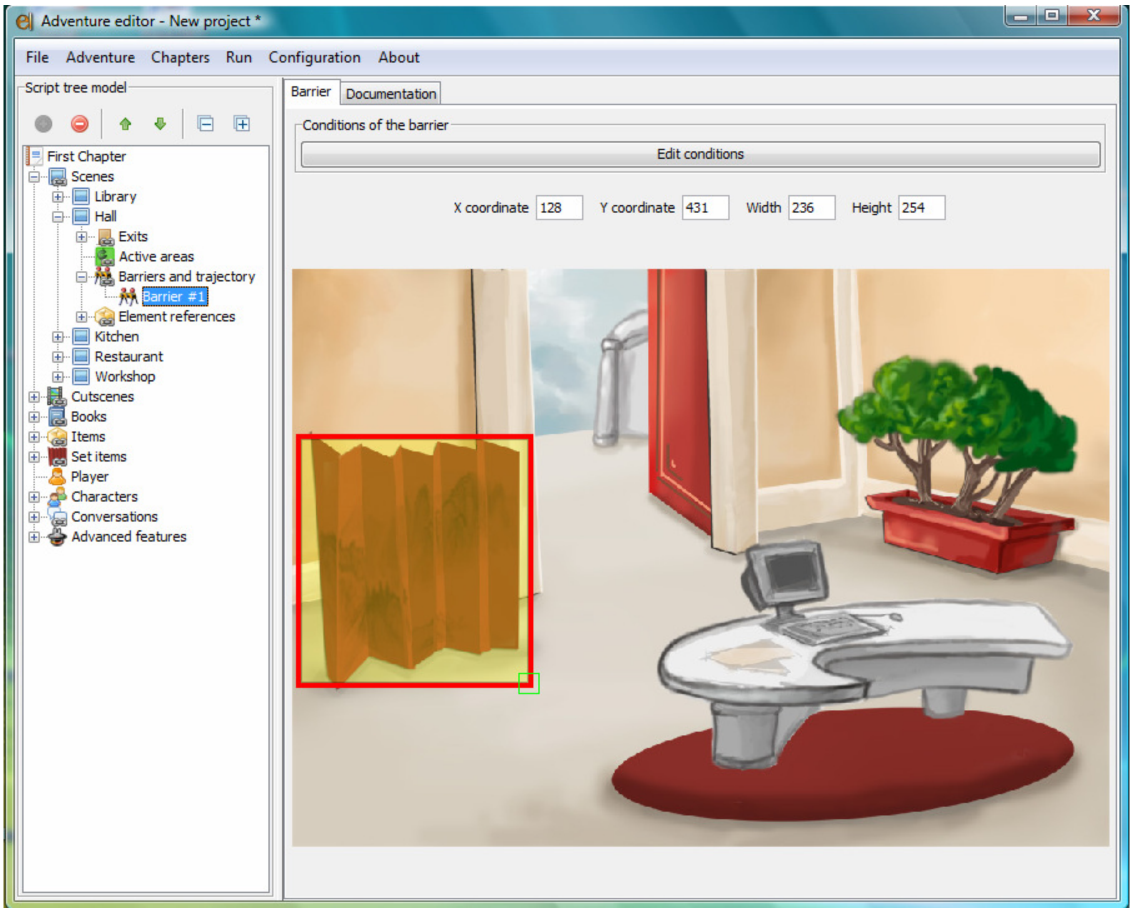
X coordinate 539 Y coordinate 397 Width 164 Height 127

3.7. Trajectories and barriers

NOTE: The barriers and trajectories are only available in the Third Person mode.

The barriers are just like the exits and active areas, rectangular regions that are defined in a scene. The barriers have a very specific functionality: stop the player (the other characters aren't affected by barriers) from going through a region. The barriers have a set of conditions, in such a way that when a barrier with its conditions active is in the way of the player while he is moving he stops right before it. If the barriers conditions are no longer satisfied, it disappears.

To edit a barrier we follow the same process as for any other element. We must first add a new barrier to the scene, and then we move and resize the barrier until it fits our needs and finally we can optionally set its conditions. A name can be given to a barrier, but it will only be used for easier identification in the editor (the same goes for the description).

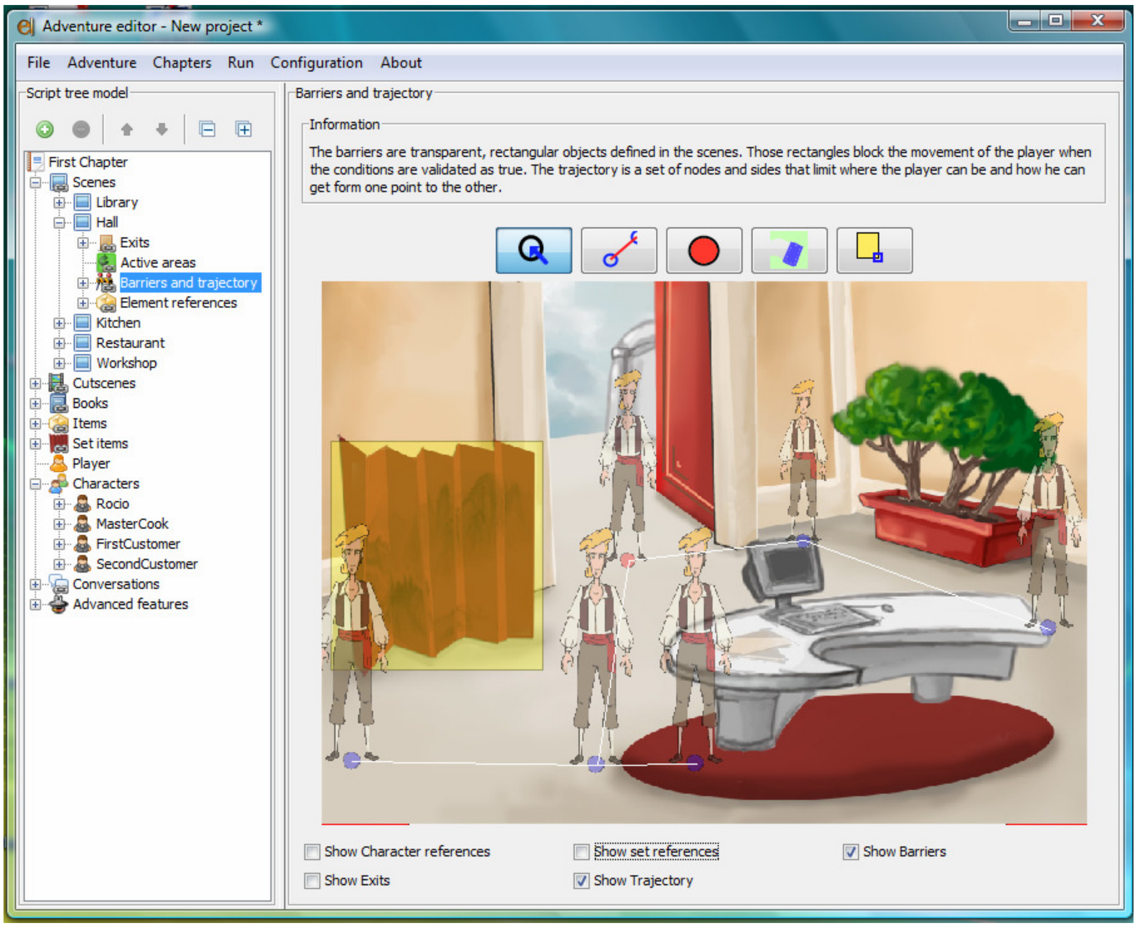


X coordinate 128 Y coordinate 431 Width 236 Height 254

Trajectories are graphs that indicate where the player **must** walk, there can only be one graph for each scene. We emphasis the word must because in a scene where the trajectories are active the player will only move along these. Both nodes and lines that go from one to the other indicate the lower central position of the player (i.e. his feet) when he moves.

Trajectories are disabled by default, they must be activated in a scene by scene basis. The checkbox to activate is named **"Use trajectories"** and can be found in the "Documentation" tab of the general scene panel.

Once the trajectories are activated, in the "Barriers and trajectories" panel we can create and edit them. To do this, we must first define the nodes (these are interest points in the path, or can be just created with the lines in mind, and we must try to limit their number as much as possible to reduce complexity) and then lines or sides between them. There can be loops, although in that case we cannot be sure that the path chosen by the player will be the best one from one point to another.



The barriers are transparent, rectangular objects defined in the scenes. Those rectangles block the movement of the player when the conditions are validated as true. The trajectory is a set of nodes and sides that limit where the player can be and how he can get form one point to the other.

☐ Show Character references ☐ Show set references ☑ Show Barriers
☐ Show Exits ☑ Show Trajectory

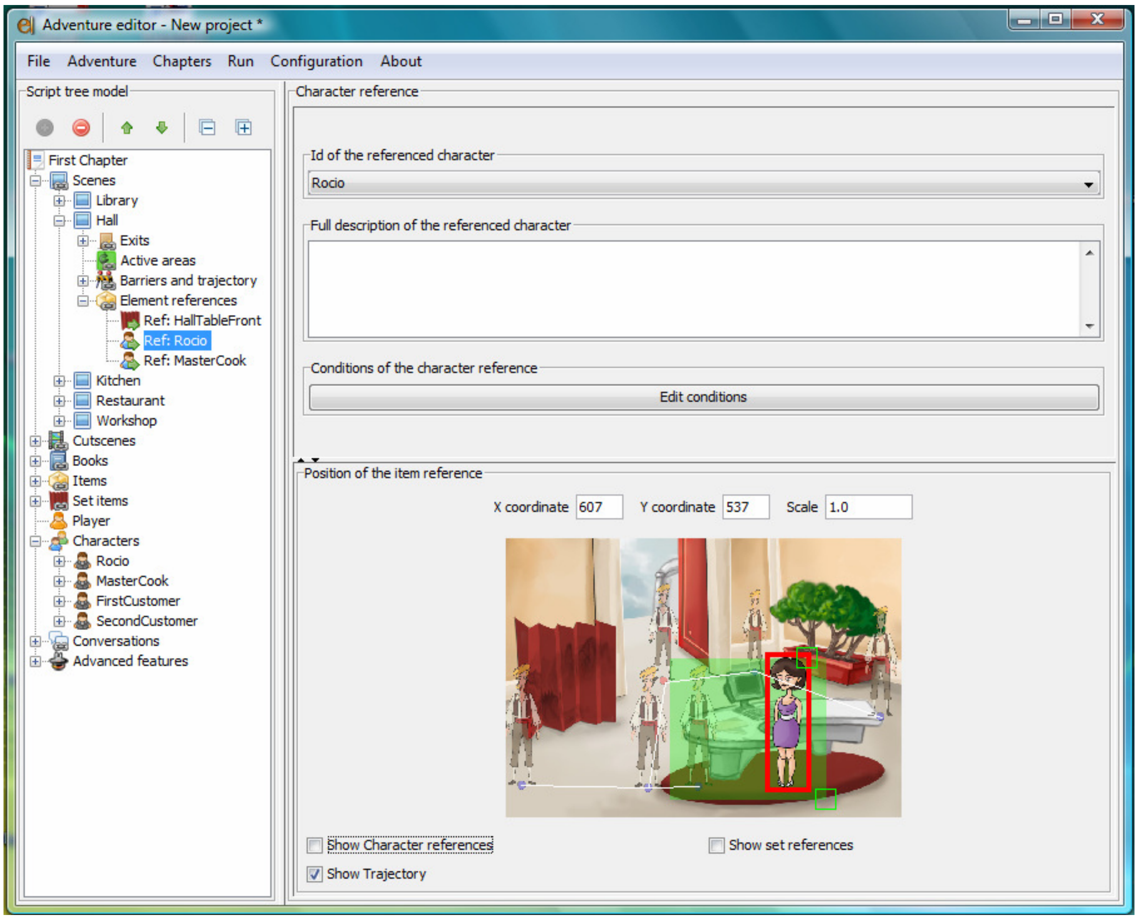
As we can see in the last figure, we have a set of tools just over the preview of the scene that we will use to edit trajectories and barriers. The tools are (in this order):

- ***"Edit nodes"***: It is used to edit nodes (select, move and rescale) as well as the creation of new nodes by clicking in an empty space.
- ***"Edit sides"***: It is used to create new sides or paths between nodes. To use this tool you must first select one node an then the other.
- ***"Select initial node"***: This tool lets us select the first node of the trajectory by clicking on it. The first node is the one over which the player appears first on the scene.
- ***"Delete tool"***: Once this tool is selected, any node or side we click will be deleted.
- ***"Edit barriers"***: It is used to edit barriers (select, move and resize)

The scale of the nodes is used to give a sense of depth to the scene, making the character appear smaller or bigger depending of the node he is at. If two nodes have different scales and there is a side from one to the other, a average of the two scales weighted by the distance to each node will be used as scale.

The barriers only affect trajectories if they are places directly over them. That is to say, in the previous screenshot the barrier would have no effect and should have been moved down if we wanted it to work.

Another thing to take into account when working with trajectories are the influence areas. These indicate from where in the trajectory we can reach an object (e.g. to grab it) and can be modified for each particular object. If the influence area of an object has no intersection with the trajectory, it will not be possible for the player to grab it if it is an item or talk to it if it is a NPC.



The influence area of the NPC "Rocio" is the green shadow around the image. The character will be able to talk to this NPC from any point in the trajectory inside this area.

3.8. Timers

Timers allow us to trigger an effects block periodically. Its usage is very simple:

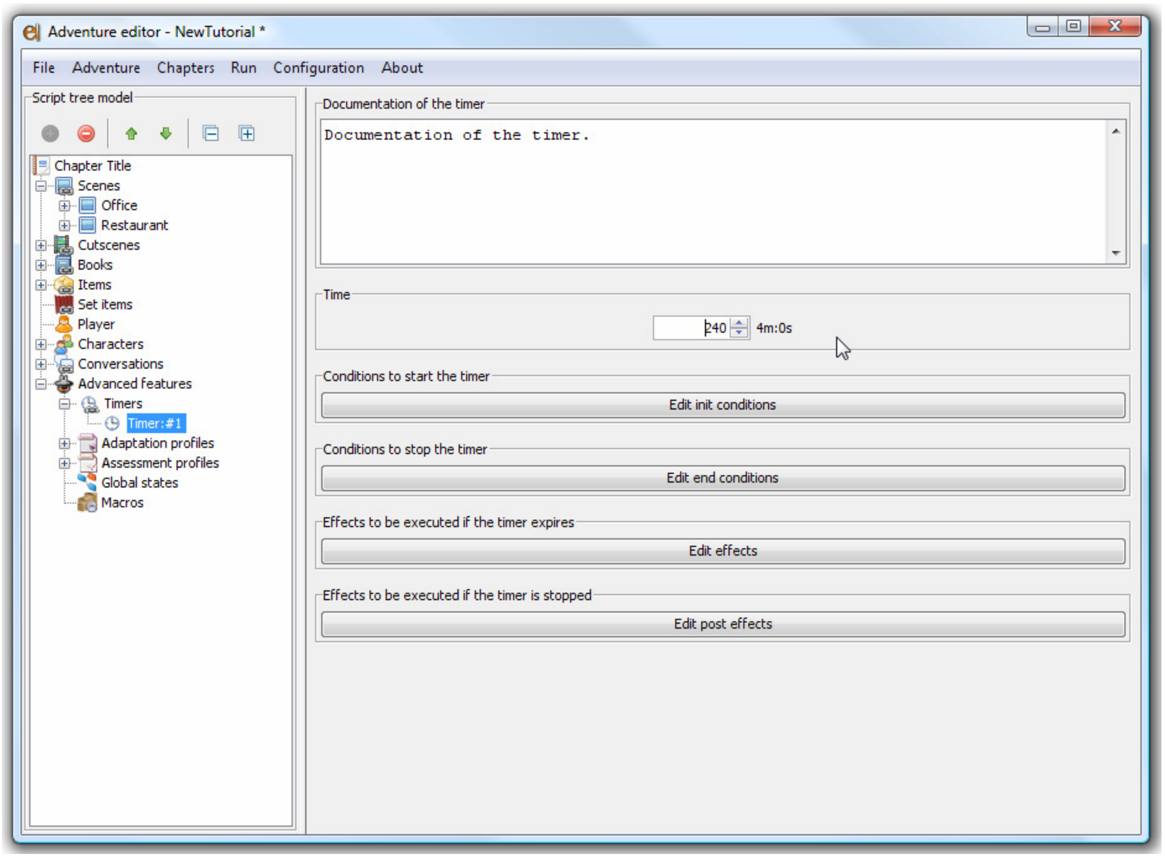
1) When the *initial conditions* are satisfied it starts counting.

2) While the *end conditions* the *effects block* is triggered each *Time* seconds. The parameter *Time* is obviously configurable.

3) Once the *end conditions* are satisfied the count ends and the *post effects* are triggered.

If we only want to get the effects block triggered once, you can just include an activate effect in the effects block that activates the flag "endTimer". Then include the condition (endTimer active) on the end conditions.

Define timers carefully. A common mistake is to define a timer with no end conditions. An empty set of conditions is always evaluated as true, so as soon as the timer is triggered the end conditions will be evaluated as true and then the timer will end without triggering any effects in the game.



Documentation of the timer

```
Documentation of the timer.
```

Time: 240 4m:0s

Conditions to start the timer: Edit init conditions

Conditions to stop the timer: Edit end conditions

Effects to be executed if the timer expires: Edit effects

Effects to be executed if the timer is stopped: Edit post effects

3.9. Custom actions

By default, the player can only use a limited number of predefined actions for each object (grab, examine and use), for each NPC (examine, talk and use-with) and for each active area (grab, examine, use and use-with). Using custom actions we can extend the way the player interacts with his surroundings by adding a completely new set of actions. This as good a time as any to remind you that, for evident reasons if you play around with the "Traditional GUI", personalized actions only work when the "Contextual GUI" is used.

Each new action needs some icons, the ones shown to the user, which must be at least a button without being pressed and a highlighted button (normally, the button pressed icon is the same as this last one).

For example, if we wanted to add a new "Paint" action, we could create a new icon with a brush like the one shown in the following figure:

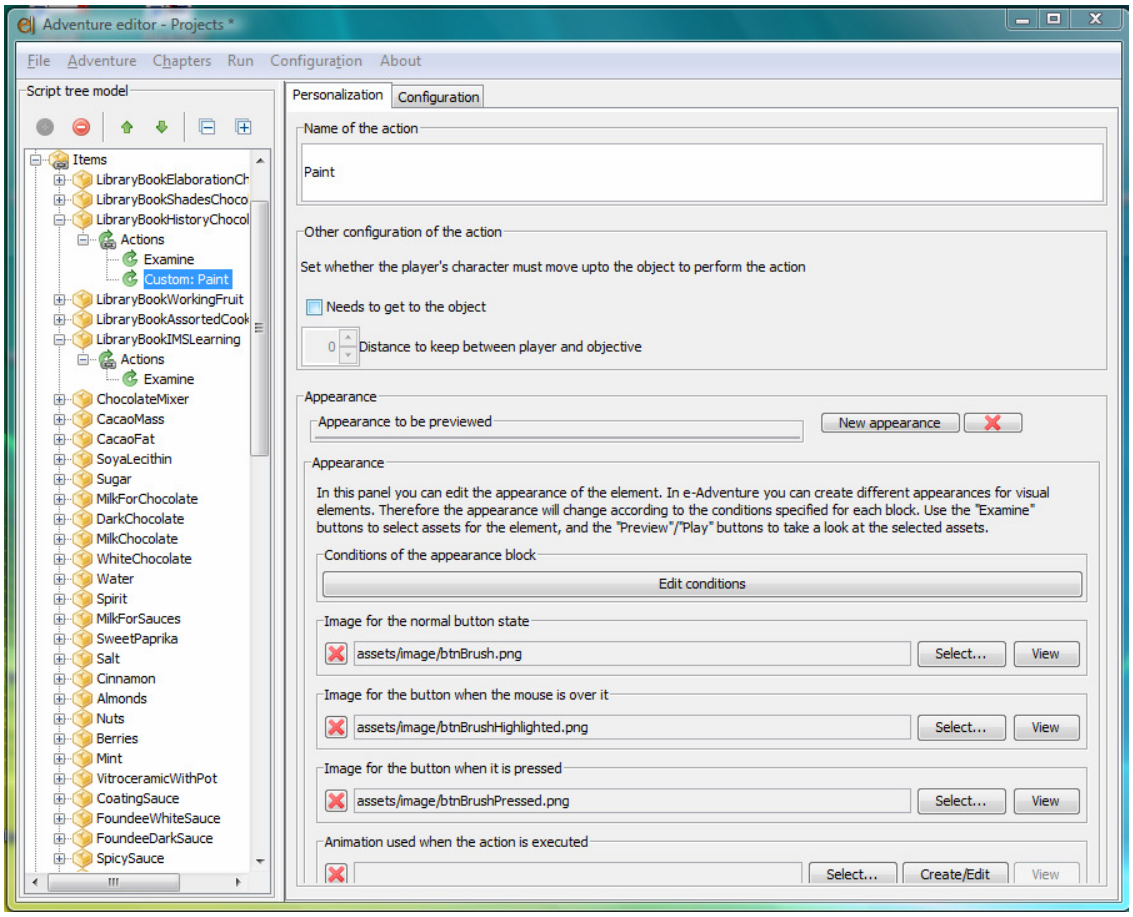
Normal icon	Highlighted icon	Pressed icon
btnBrush.png	btnBrushHighlighted.png	btnBrushPressed.png

We must then add the new action to the desired object, NPC or active area. When we choose to *Add "Custom" action* the system we ask for a name. After we give the action a name, we will be asked for the type of action we what to create:

- ***Action***: these only need the object

- ***Interaction***: these need another object, NPC or active area to work, just like use-with actions

In the example, we chose "Paint" as the name for the new action and "Action" as its type. We get the following result:



Adventure editor - Projects *

File Adventure Chapters Run Configuration About

Script tree model

Items
LibraryBookElaborationCh
LibraryBookShadesChoco
LibraryBookHistoryChocol
Actions
Examine
Custom: Paint
LibraryBookWorkingFruit
LibraryBookAssortedCook
LibraryBookIMSLearning
Actions
Examine
ChocolateMixer
CacaoMass
CacaoFat
SoyaLecithin
Sugar
MilkForChocolate
DarkChocolate
MilkChocolate
WhiteChocolate
Water
Spirit
MilkForSauces
SweetPaprika
Salt
Cinnamon
Almonds
Nuts
Berries
Mint
VitroceramicWithPot
CoatingSauce
FoundeeWhiteSauce
FoundeeDarkSauce
SpicySauce

Personalization | Configuration

Name of the action
Paint

Other configuration of the action
Set whether the player's character must move upto the object to perform the action
Needs to get to the object
0 Distance to keep between player and objective

Appearance
Appearance to be previewed | New appearance

Appearance
In this panel you can edit the appearance of the element. In e-Adventure you can create different appearances for visual elements. Therefore the appearance will change according to the conditions specified for each block. Use the "Examine" buttons to select assets for the element, and the "Preview"/"Play" buttons to take a look at the selected assets.

Conditions of the appearance block
Edit conditions

Image for the normal button state
assets/image/btnBrush.png | Select... | View

Image for the button when the mouse is over it
assets/image/btnBrushHighlighted.png | Select... | View

Image for the button when it is pressed
assets/image/btnBrushPressed.png | Select... | View

Animation used when the action is executed
Select... | Create/Edit | View

As we can see, we have already chosen the images for the buttons, using the ones shown before. In the "Configuration" tab will find the options common to every action, information about these options can be found in the actions section of this manual.

The option "Needs to get to object" allows for different behaviors for the actions. We can create actions that can be taken from any player position (e.g. "Shout" action) and other actions that need the player to get close to the object (like the previously described "Paint" action). If this option is selected, the action can only be taken if there are no barriers or other obstacles that prevent the player from reaching the object.

It is also possible to change the animation shown will the player is making the action. If no animation is selected one will be used by default, but this is not recommended given that using a custom animation (a different one when the player is painting from the one used when the player is grabbing an object) adds expressivity to the game.

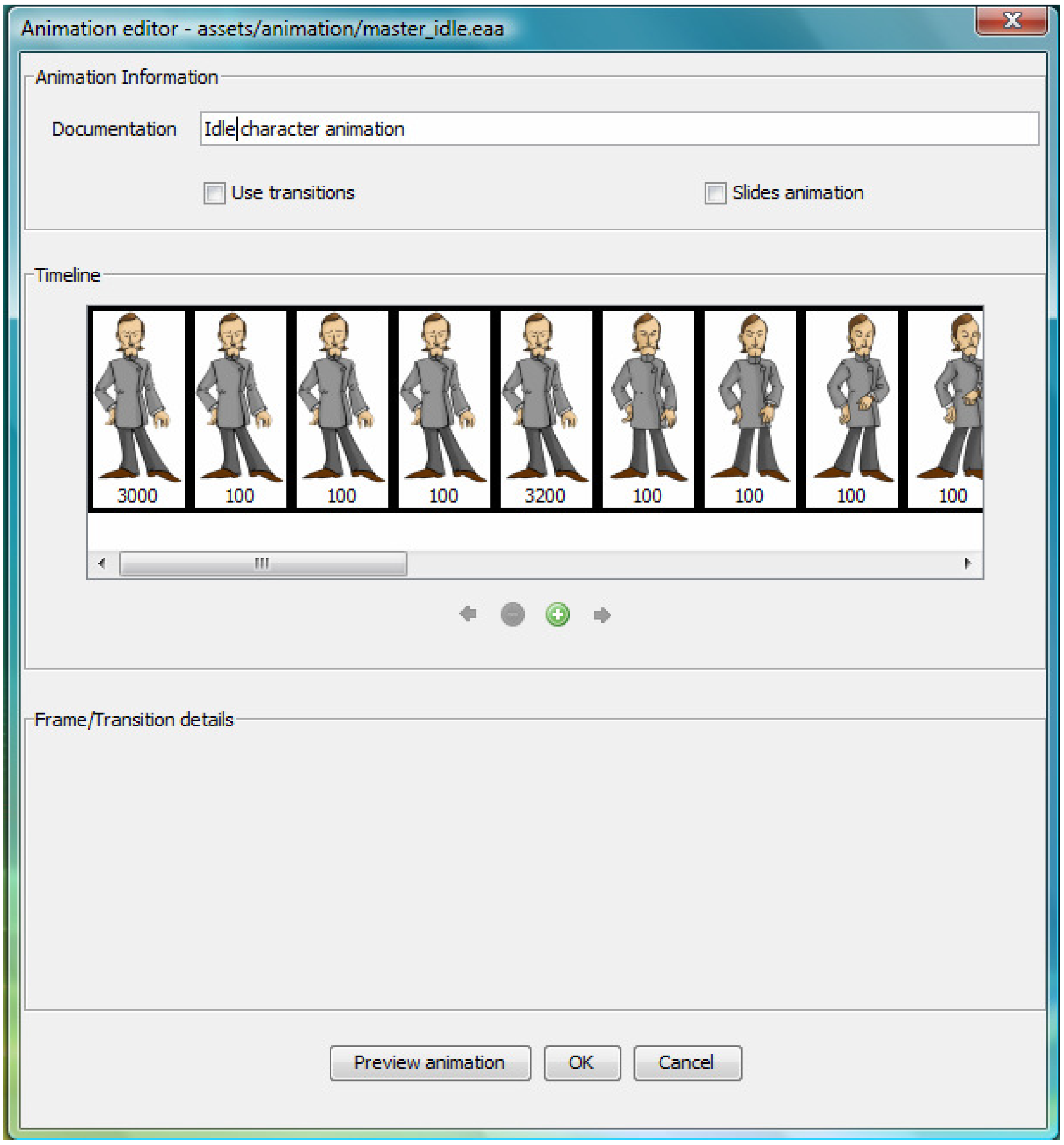
3.10. Advanced asset edition: Animations and HTML editors.

In this version of the editor, two new tools are added that allow a more detailed edition of the resources without leaving the editor: the animation editor and the HTML editor. Both tools make it easier to use the full capacity of expression of the platform, making it easier to create complex games without the need for external tools.

3.10.1. Animation Editor.

The animation creates files with the ".eaa" extension. The same editor makes it easy to import animations in the old format and does this automatically. The new animations are used for the characters as well as for the slideshows, making it possible to use different durations for each frame of an animation of slide in a slideshow as well as the use of transition effects from one to the other.

To use this editor, we must click in the button called "Create/Edit" that is placed in any animation resource needed for the game. It looks like this:


Animation editor - assets/animation/master_idle.eaa

Animation Information

Documentation: Idle character animation

☐ Use transitions ☐ Slides animation

Timeline

3000	100	100	100	3200	100	100	100	100

Frame/Transition details

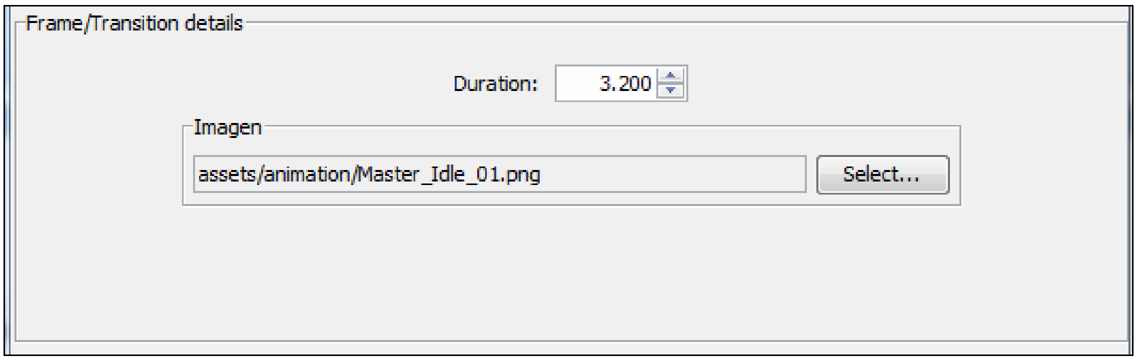
The documentation field is used to make it easier to identify the animation in the editor but is not used in the engine. Below we have the following checkboxes:

- *"Use transitions"*: It activates the transition effects in between animation frames (or slideshow slides)
- *"Slides animation"*: Must only be used for slideshows, it allows the possibility to set a frame (or slide) to be shown until the player chooses to go to the next one.

In the timeline we can see for each frame, beneath the image, the time that is shown. That way if a frame has 3000 written on it, it means that it will appear for 3000 milliseconds before the next one is shown.

The buttons that appear under the timeline are used to add a new frame, remove the selected frame, move the selected frame forward in the timeline or move the selected frame backwards in the timeline.

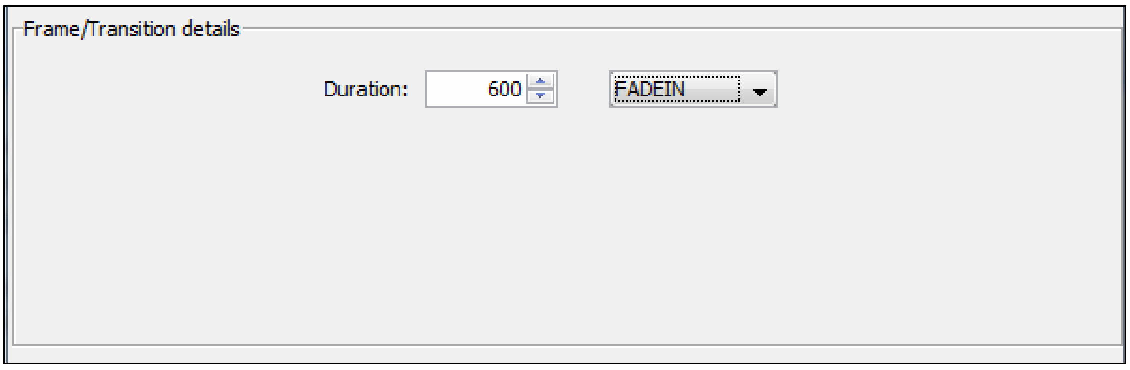
The properties that can be edited for a frame, that appear in the lower part of the editor when one is selected, are the following:



Duration: 3.200
Imagen: assets/animation/Master_Idle_01.png

As we can see, it is possible to change the duration of a frame as well as its image. When we are editing a slide animation, we have a checkbox to choose if the slide is automatically changed when it's time passes or if it must wait for the user to ask for that.

If we decide to use transitions, these appear as a square between each frame. Transitions have a duration too, as well as an effect. The panel for the edition of a transition is as follows:



Duration: 600 FADEIN

Where we can change the duration as well as the effect that we can choose from:

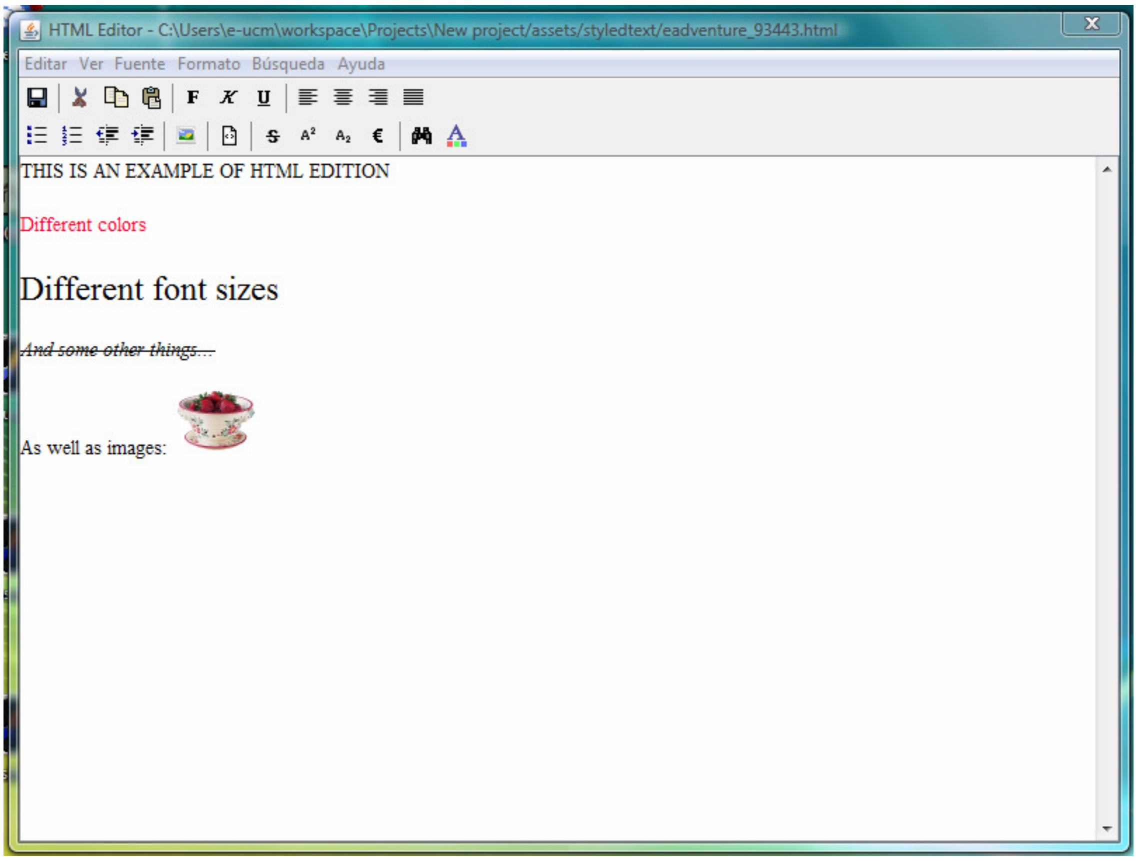
- **NONE:** No effect is used for this transition, the previous image will appear until the time is up
- **FADEIN:** The previous image fades out as the new image fades in. The speed in which the images change depends on the duration
- **HORIZONTAL:** The previous image is moved from left to right as the new image appears in the same way.
- **VERTICAL:** The previous image is moved from top to bottom as the new image appears in the same way.

At any time while we are editing an animation we can choose to use the "Preview animation" button to see the result we are achieving. The "OK" button saves the changes and the "CANCEL" button undoes any changes in the animation.

3.10.2. HTML editor.

This editor can be used when creating HTML Books or "Formatted text" books. If you decide to use a local resource for a page of the book, this can be created directly in the editor with the "Create" button or can be imported as an html file created somewhere else. Once a local resource is selected, the editor is easily reached by the "Edit" button. As we already say before in this manual, HTML books have some problems with metadata so it is recommended no to add this kind of tags to the books.

The HTML Editor is a fully functional WYSIWYG (What You See Is What You Get) editor. Inside the editor you can add new resources such as images to the document as well as change the fonts of the text (kind, size, italics, etc.).


HTML Editor - C:\Users\e-ucm\workspace\Projects\New project/assets/styledtext/eadventure_93443.html
Editar Ver Fuente Formato Búsqueda Ayuda

THIS IS AN EXAMPLE OF HTML EDITION

Different colors

Different font sizes

~~*And some other things...*~~

As well as images:

Once the changes are saved and the editor is closed, the result can be seen inside the "Book preview".

NOTE: It is strongly recommended not to try to adjust the margins directly in the editor but to use the tool created specifically for that end. Said tool can be accessed clicking in the "Edit margins" button in the page edition panel.

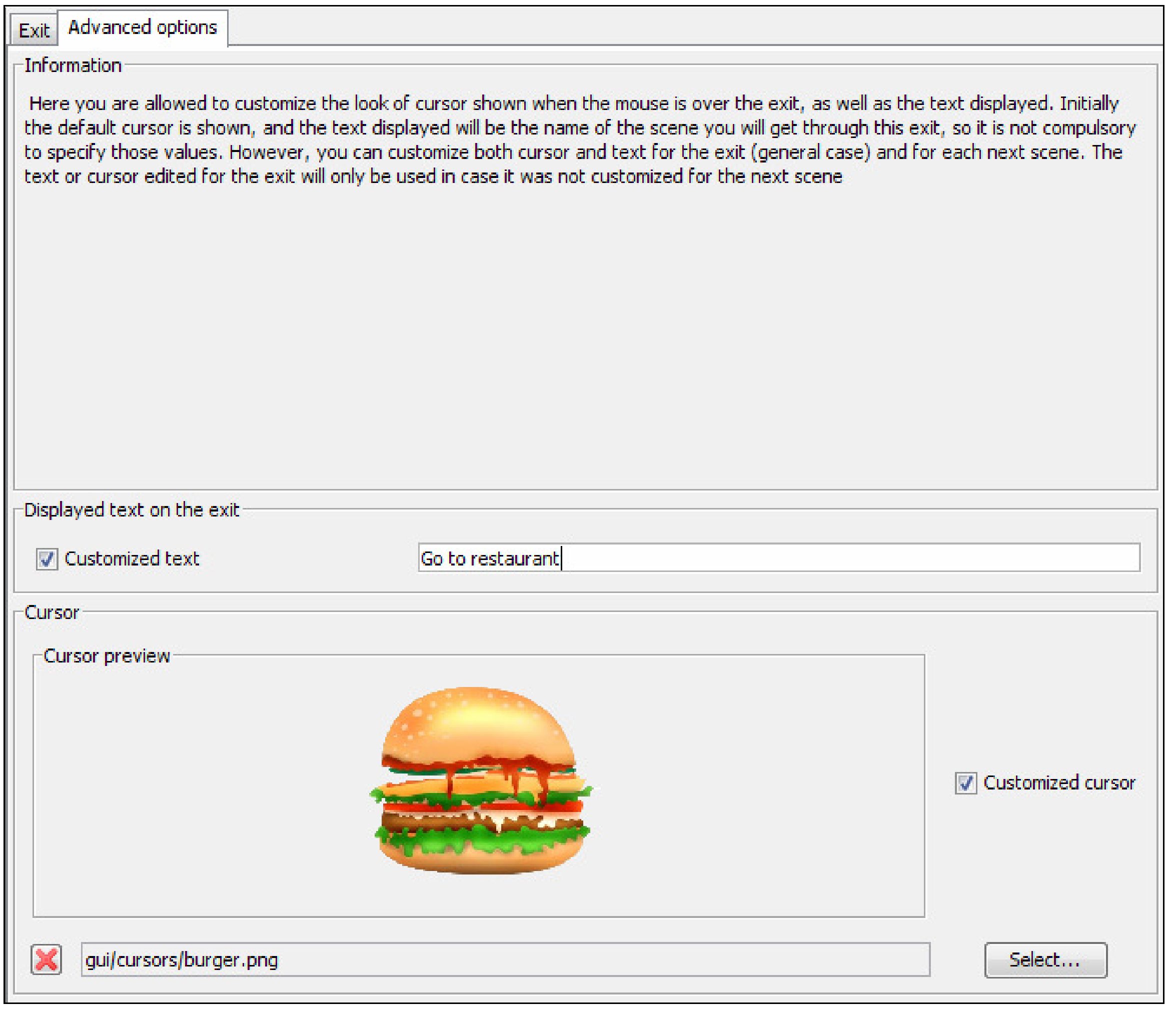
3.11. Advanced settings for exits.

The exits and the "Next scene" structures have a second tab "Advanced features" that is to be explored yet in this manual.

This tab allows us to configure two parameters: The text to display when the cursor is over an exit, and the image of the cursor as well.

The usage of custom cursors is as follows:

In first place the "Next scene" structure is checked. In case it contains text label/cursor personalization those values are retrieved and used when rendering the scene. If not, the whole "Exit" is checked. If the game engine does not find personalized data there it will next check the custom cursors of the adventure (see next section). At last, if no custom cursors are found at all the game engine will use the default cursor and the name of the scene to go.



Here you are allowed to customize the look of cursor shown when the mouse is over the exit, as well as the text displayed. Initially the default cursor is shown, and the text displayed will be the name of the scene you will get through this exit, so it is not compulsory to specify those values. However, you can customize both cursor and text for the exit (general case) and for each next scene. The text or cursor edited for the exit will only be used in case it was not customized for the next scene

Displayed text on the exit

[✓] Customized text: Go to restaurant

Cursor

Cursor preview

[✓] Customized cursor

gui/cursors/burger.png

3.12. Other options.

In this section we describe how to use the menus of the menu bar to customize further options of the <e-Adventure> games.

File menu

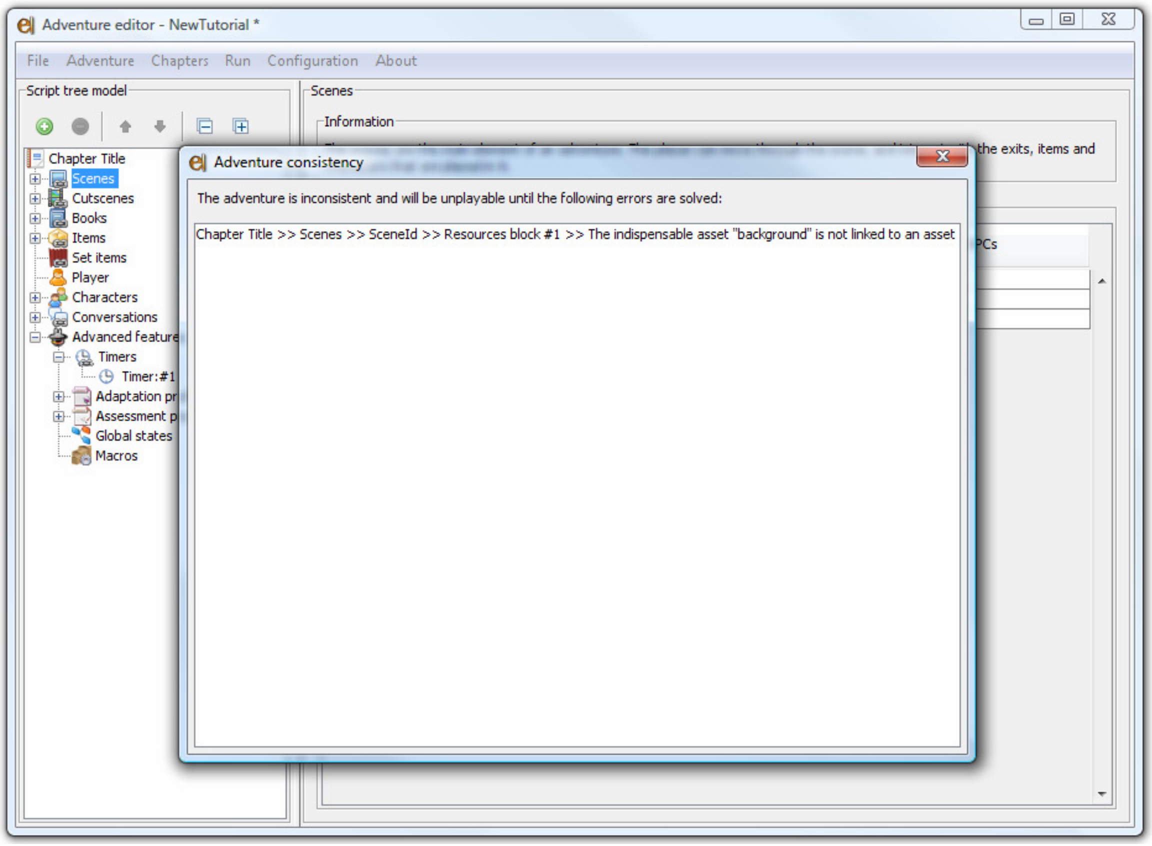
This menu provides conventional options to save, load and create new <e-Adventure> files.

Adventure menu

This menu provides options for the edition of general data of the adventure. Those options are next described.

Check adventure consistency

This option of the menu generates a report with all the incidences that will prevent the game to be executed. For instance, an aspect that will lead to invalidation of the file is to keep a scene with no background.

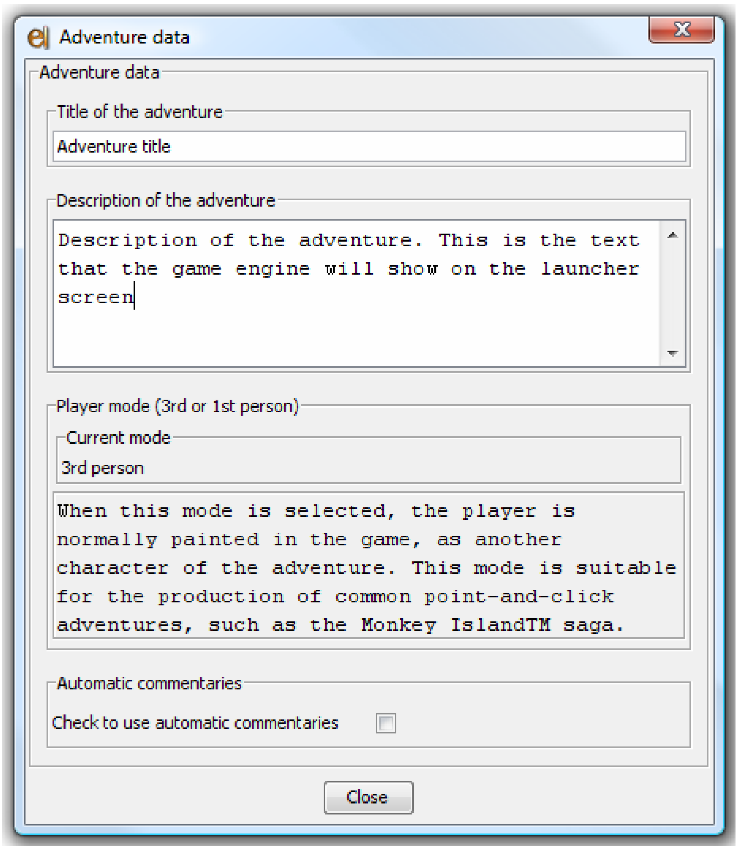


The adventure is inconsistent and will be unplayable until the following errors are solved:

Chapter Title >> Scenes >> SceneId >> Resources block #1 >> The indispensable asset "background" is not linked to an asset

Edit adventure data

Provides a dialog for the edition of general data of the game. That is a title and a description. Besides, it displays information about the current mode of the game (player visible or transparent). Finally it allows users to deactivate default commentaries in the game when an action is not allowed.


Adventure data

Adventure data

Title of the adventure: Adventure title

Description of the adventure: Description of the adventure. This is the text that the game engine will show on the launcher screen

Player mode (3rd or 1st person)

Current mode: 3rd person

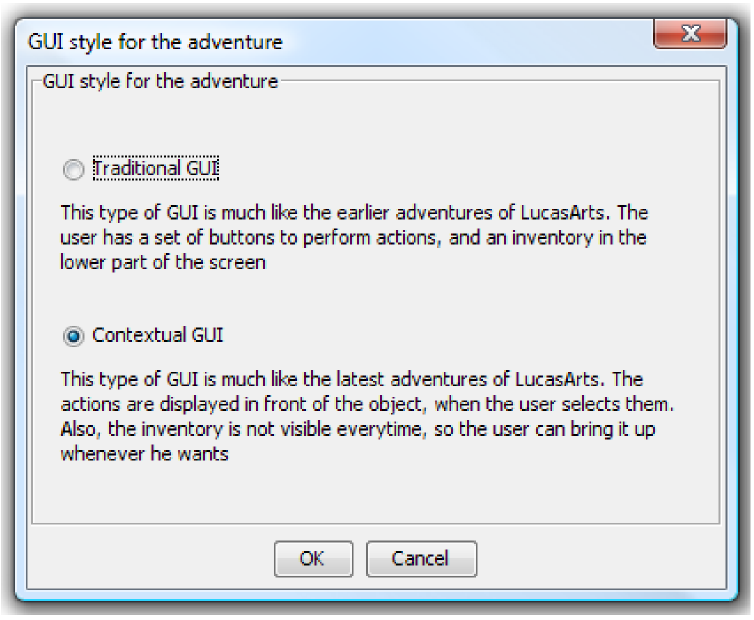
When this mode is selected, the player is normally painted in the game, as another character of the adventure. This mode is suitable for the production of common point-and-click adventures, such as the Monkey IslandTM saga.

Automatic commentaries

Check to use automatic commentaries ☐

Visualization > Select GUI Style

In <e-Adventure> the game can be presented using two different **Graphic User Interfaces (GUI)**: Traditional GUI and Contextual GUI. By default the selected is the Traditional GUI. The choice of the GUI will change how the game is perceived by the player and how the interaction flows.


GUI style for the adventure

GUI style for the adventure

○ Traditional GUI

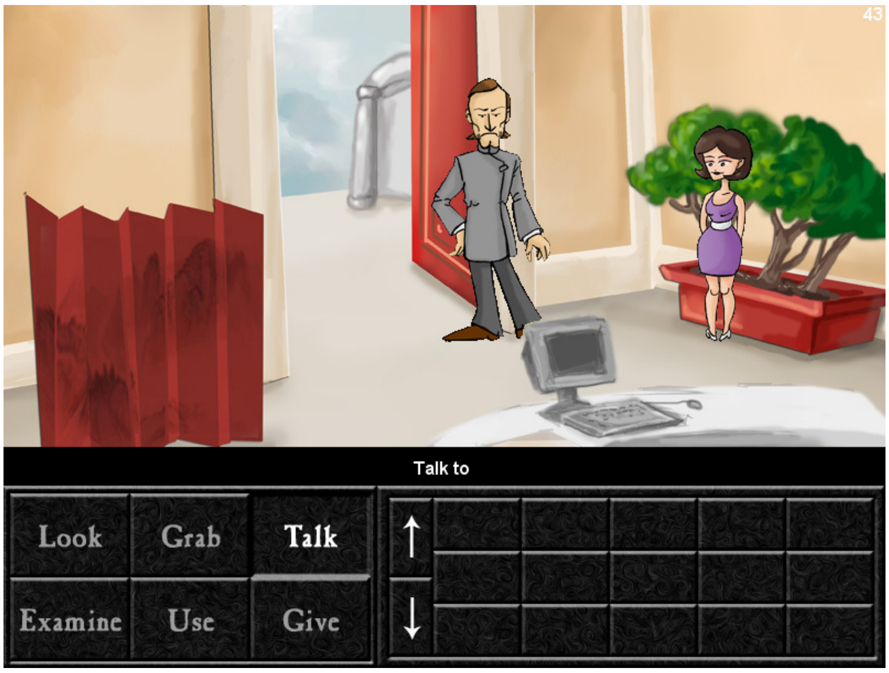
This type of GUI is much like the earlier adventures of LucasArts. The user has a set of buttons to perform actions, and an inventory in the lower part of the screen

● Contextual GUI

This type of GUI is much like the latest adventures of LucasArts. The actions are displayed in front of the object, when the user selects them. Also, the inventory is not visible everytime, so the user can bring it up whenever he wants

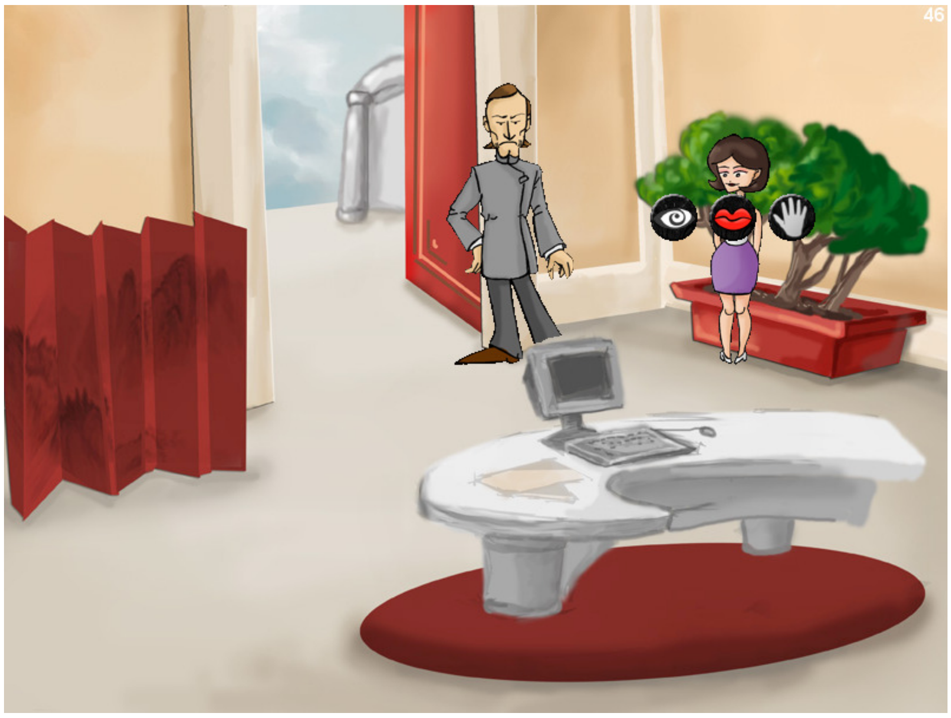
Next you will see an example of the same game using Traditional GUI first and then Contextual GUI.

Traditional GUI



This kind of GUI is very similar to the first point-and-click adventure games of *LuscasArts*^{TM}. At any moment of the game there is a panel covering the bottom of the screen where you can select what you want to do (actions) using the buttons on the left and your inventory is presented on the right.

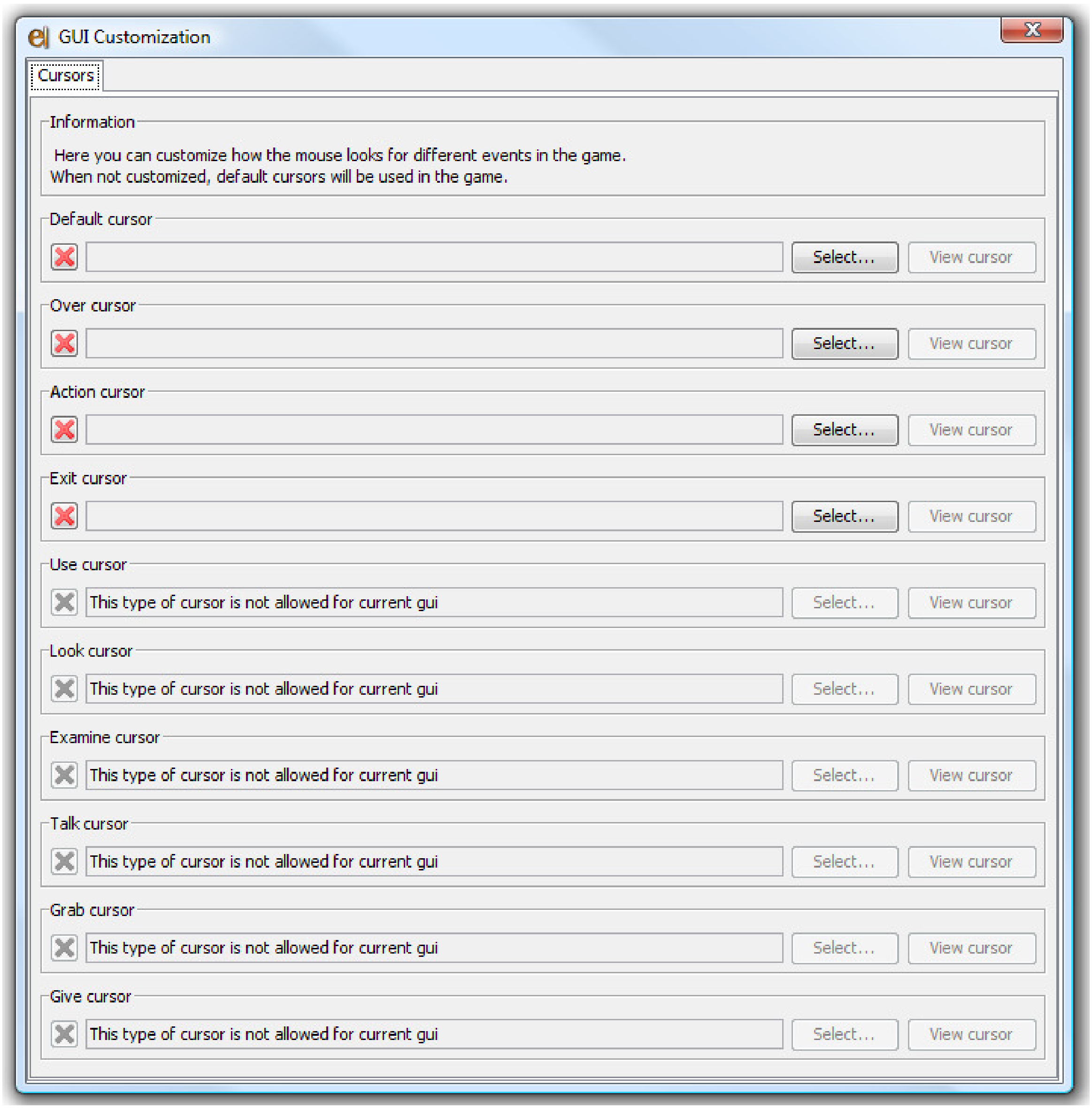
Contextual GUI



As you see, in this case the GUI panel is not always shown. On the contrary, if you point your mouse on an item or character you will see its name and the cursor will change. Then you know there is something you can do with it. If you click on it the different options will be displayed there as small icon-buttons. The inventory is only displayed when you point the mouse over the top or the bottom of the screen.

Visualization > GUI Customization

<e-Adventure> lets you customize some aspects of the graphical interface. For this release you can only customize the cursors used in the game, but soon new functionalities will be added.


GUI Customization

Cursors

Information

Here you can customize how the mouse looks for different events in the game.
When not customized, default cursors will be used in the game.

Default cursor		Select...	View cursor
Over cursor		Select...	View cursor
Action cursor		Select...	View cursor
Exit cursor		Select...	View cursor
Use cursor	This type of cursor is not allowed for current gui	Select...	View cursor
Look cursor	This type of cursor is not allowed for current gui	Select...	View cursor
Examine cursor	This type of cursor is not allowed for current gui	Select...	View cursor
Talk cursor	This type of cursor is not allowed for current gui	Select...	View cursor
Grab cursor	This type of cursor is not allowed for current gui	Select...	View cursor
Give cursor	This type of cursor is not allowed for current gui	Select...	View cursor

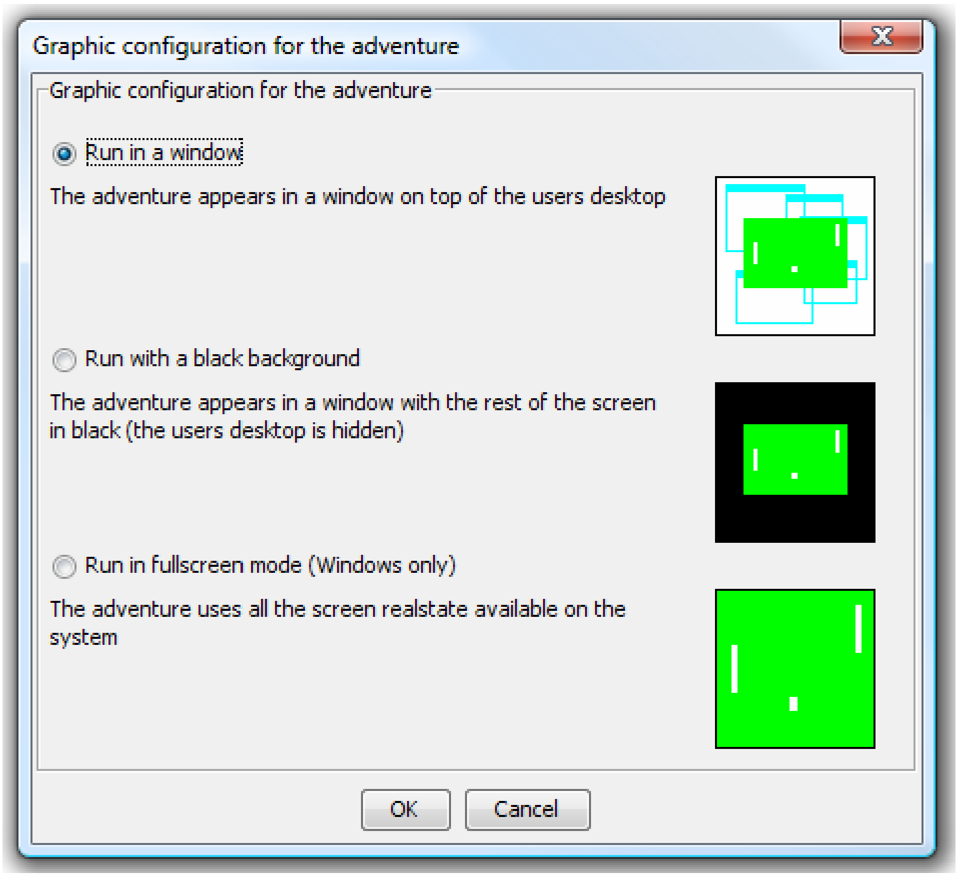
Just customize a cursor by clicking on its "Select..:" button. Choose an image and it will be used as cursor. Note that not all the cursors are available for both GUIs.

Visualization > Select Graphics Configuration

This dialog allows us to choose the graphic mode of the game. In <e-Adventure> there are three supported settings. The first one, which is the default option, runs the game in a 800x600 pixels window which is centred on the screen of the user (***"Run in a window"***).

The second mode (***"Run with a black background"***) adds a black background to the window which hides the desktop. This mode is especially adequate to avoid distractions of the students.

The third mode is only supported on Windows systems, and runs the game in full screen mode. Nonetheless the resolution will be 800x600 anyway, so its use is not recommended in screens with a really high resolution.


Graphic configuration for the adventure

Graphic configuration for the adventure

[✓ Run in a window]

The adventure appears in a window on top of the users desktop

Run with a black background

The adventure appears in a window with the rest of the screen in black (the users desktop is hidden)

Run in fullscreen mode (Windows only)

The adventure uses all the screen realstate available on the system

Change player mode (Convert to Third person mode / Convert to First person mode)

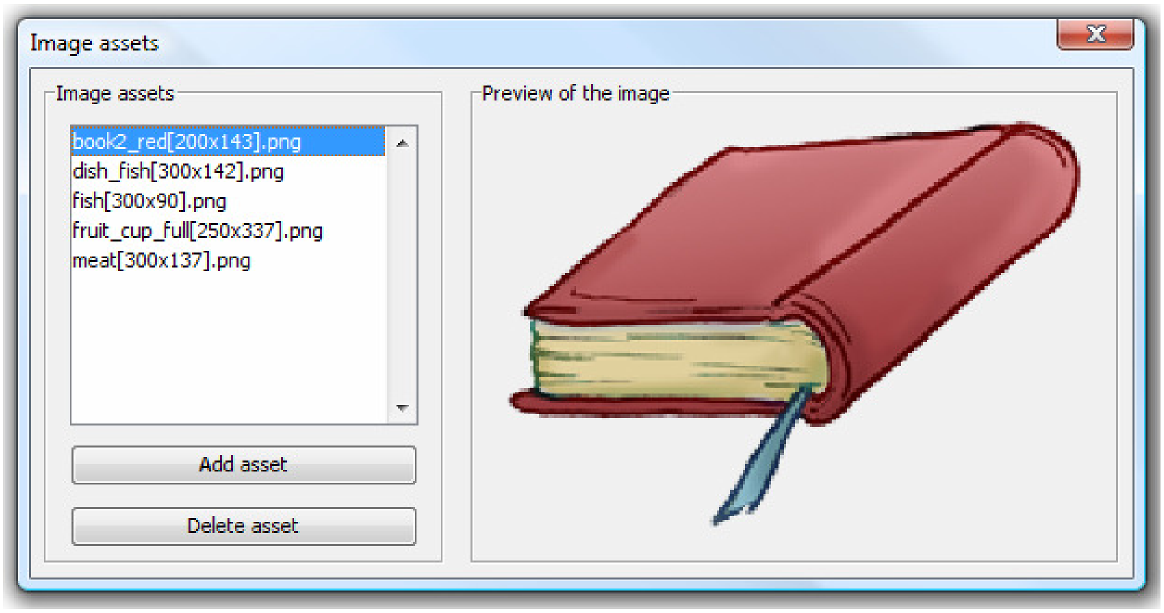
As you know, <e-Adventure> supports two different kinds of games categorized according to how the player is perceived. Using the mode "Third person mode" you will create adventures where the player is shown normally, as in most conventional point-and-click games like, for instance, the *MonkeyIsland*^{TM} saga.

On the other hand, using the mode "First person mode" you will create adventures where the player is not shown. The games created under this mode will look like the *Myst*^{TM} adventure saga, and specially devised for photo-realism works.

You can change the current mode of an adventure game by clicking on this option of the Adventure menu. Do not be afraid to do so, as this process is completely revertible.

Assets management

There you have options to add and delete assets of the different categories directly. Remember those are automatically added to the project folder, which contains absolutely everything in the game, but they are never deleted. For instance, if you click on "Manage image assets" you will be prompted with a dialog like this, where you can preview the items of the category and delete them or add new ones. If you add some using this facility you will be able to select them in the editor when a chooser is prompted by clicking on the project folder on the bottom left (only on Windows).



Chapters menu

Use this chapter to change the current chapter you are editing, add chapters, delete chapters or edit the flags involved in the chapter.

Run Menu

This menu is a shortcut for the execution of the games without the game engine. You can run the <e-Adventure> games by clicking on the "Normal Run" option of the menu or pressing the combination of keys Ctrl+R.

Configuration menu

This menu allows you to customize some aspects of how the editor looks like. Those are:

Show item references by default

If selected, when you see the preview of a scene you will see all the item references.

Show character references by default

If selected, when you see the preview of a scene you will see all the character references.

Show set item references

If selected, when you see the preview of a scene you will see al the set item references.

Show start dialog

If you deselect this option the next time you run the editor.

Set Language

Use this menu to change the language of the editor (English or Spanish only).